UNIVERSITY OF THE AEGEAN

DEPARTMENT OF FINANCIAL & MANAGEMENT ENGINEERING

SOLVING THE DYNAMIC VEHICLE ROUTING PROBLEM WITH MIXED BACKHAULS THROUGH RE-OPTIMIZATION

A Ph.D. dissertation presented

by

GEORGIOS NINIKAS

Chios, Greece

December, 2014

UNIVERSITY OF THE AEGEAN
SCHOOL OF BUSINESS
DEPARTMENT OF FINANCIAL & MANAGEMENT ENGINEERING

SOLVING THE DYNAMIC VEHICLE ROUTING PROBLEM WITH MIXED BACKHAULS THROUGH RE- OPTIMIZATION

A Dissertation Presented by

GEORGIOS NINIKAS

in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

DISSERTATION COMMITTEE

IOANNIS MINIS, PROFESSOR (Supervisor)
Department of Financial and Management Engineering, University of the Aegean

EPAMINONDAS KIRIAKIDIS, PROFESSOR (Advisory Committee Member)
Department of Statistics, Athens University of Economics and Business

AGAPIOS PLATIS, ASSOCIATE PROFESSOR (Advisory Committee Member)
Department of Financial and Management Engineering, University of the Aegean

GEORGIOS DOUNIAS, PROFESSOR (Examination Committee Member)
Department of Financial and Management Engineering, University of the Aegean

DIMITRIOS EMIRIS, PROFESSOR (Examination Committee Member)
Department of Industrial Management and Technology, University of Pireaus

DIMITRIOS KOULOURIOTIS, ASSOCIATE PROFESSOR (Examination Committee Member)
Department of Production and Management Engineering, Democritus University of Thrace

CHRISTOS TARANTILIS, PROFESSOR (Examination Committee Member)
Department of Management Science and Technology, Athens University of Economics and Business

Chios, Greece
December, 2014

ACKNOWLEDGEMENTS

I would like to address my first and deepest appreciation to Professor Ioannis Minis for his valuable supervision, guidance and support throughout the PhD years. I will always be grateful for his strong belief in me and his generous mentorship for the past 10 years that has significantly contributed to my professional and personal evolvement.

Secondly, special thanks to my very close friend and colleague Dr. Theodore Athanasopoulos for his continuous support, encouragement and patience for all the endless discussions throughout the dissertation. I feel very grateful having him as a close business partner for the past 10 years. Furthermore, I would like to express my appreciation to Dr. Georgios Dikas for the valuable support with bright ideas and solutions on challenging modeling aspects in this research. Along with the aforementioned colleagues, I would also like to acknowledge the support of the entire staff in the DeOPSys Lab of the Financial and Management Engineering Department of the University of the Aegean and especially for all the resources made available to me throughout my dissertation years.

Last but not least, this PhD would never be possible without the selfless love, moral support and encouragement from my parents, Achilleas and Angeliki and my brother and best friend Kostas, who all stood by me during the difficult times of this venture. Finally, special thanks have to go to all my close friends and especially to Stefanos Passadis, for strongly believing in me and making some of the difficult times easier and funnier.

to my parents and
my brother

EXTENDED SUMMARY IN GREEK (ΣΥΝΟΨΗ ΔΙΑΤΡΙΒΗΣ)

Στη παρούσα διατριβή διερευνάται το Πρόβλημα Δυναμικής Δρομολόγησης Οχημάτων με Παραλαβές (ΠΔΔΟΠ). Στόχος του προβλήματος είναι η βέλτιστη ανάθεση δυναμικών απαιτήσεων παραλαβών που λαμβάνονται σε πραγματικό χρόνο σε στόλο οχημάτων που εκτελεί προκαθορισμένα δρομολόγια «στατικών» παραδόσεων. Το πρόβλημα ενσωμάτωσης των δυναμικών απαιτήσεων αντιμετωπίζεται με περιοδική αναδρομολόγηση. Για την επίλυση του προβλήματος αναδρομολόγησης, προτείνεται νέο μαθηματικό μοντέλο, καθώς και νέα προσέγγιση βέλτιστης επίλυσης μέσω της μεθόδου Branch-and-Price (B&P). Για την επίλυση απαιτητικών προβλημάτων (π.χ. χωρίς χρονικά παράθυρα), προτείνεται καινοτόμος ευρετική μέθοδος παρεμβολής (insertion heuristic) που βασίζεται στη μέθοδο Δυναμικής Δημιουργίας Μεταβλητών (ΔΔΜ ή Column Generation) και παρέχει αποτελεσματικές λύσεις σε σύντομο υπολογιστικό χρόνο με μικρή απόκλιση από τη βέλτιστη.

Χρησιμοποιώντας τη προαναφερόμενη προσέγγιση, η διατριβή επικεντρώνεται επίσης στη διαδικασία αναδρομολόγησης, που αποτελείται από: α) την πολιτική αναδρομολόγησης (συχνότητα), και β) τη τακτική υλοποίησης. Η τελευταία σχετίζεται με το τμήμα του δρομολογίου που κοινοποιείται στον οδηγό προς εκτέλεση. Παρουσιάζονται και αναλύονται πρακτικές στρατηγικές αναδρομολόγησης (συνδυασμός πολιτικής και τακτικής) μέσω εκτενούς πειραματικής διερεύνησης, αρχικά θεωρώντας απεριόριστο στόλο οχημάτων διαθέσιμο με στόχο μόνο την ελαχιστοποίηση του κόστους. Βάσει των αποτελεσμάτων, προτείνονται οδηγίες για την υιοθέτηση της καταλληλότερης στρατηγικής αναδρομολόγησης ανάλογα με τα εκάστοτε χαρακτηριστικά του περιβάλλοντος της εφοδιαστικής αλυσίδας (π.χ. γεωγραφική κατανομή, χρονικά παράθυρα πελατών, δυναμικότητα, κλπ.).

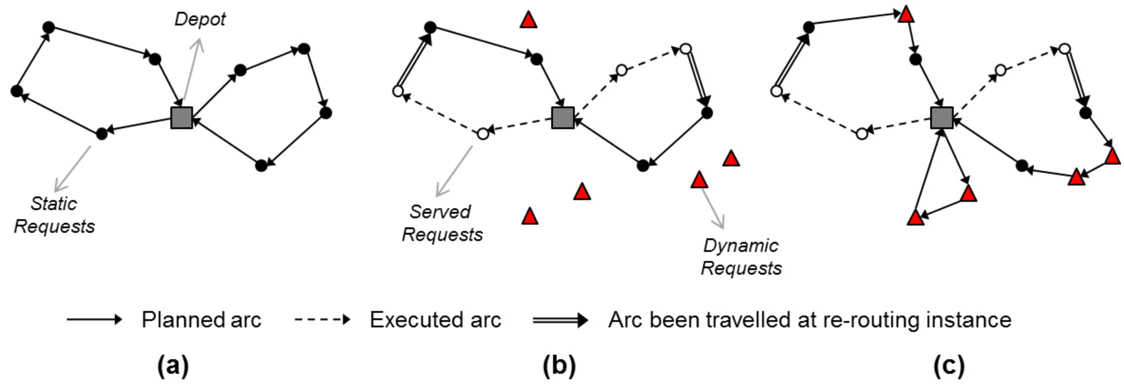
Ακολούθως, μελετάται η περίπτωση περιορισμένου στόλου οχημάτων στην οποία μόνο ένα μέρος των δυναμικών απαιτήσεων μπορεί να εξυπηρετηθεί. Για την αντιμετώπιση του προβλήματος, προτείνονται οι απαραίτητες αλλαγές τόσο στο μοντέλο ΠΔΔΟΠ, όσο και στη μέθοδο επίλυσης. Όσον αφορά το πρόβλημα αναδρομολόγησης, χρησιμοποιούμε αρχικά μία συμβατική αντικειμενική συνάρτηση, η οποία προσπαθεί να μεγιστοποιήσει την εξυπηρέτηση πελατών. Για την περίπτωση αυτή, υποδεικνύουμε μέσω πειραματικής διερεύνησης πως οι στρατηγικές αναδρομολόγησης παρουσιάζουν παρόμοια συμπεριφορά με τη περίπτωση που η διαθεσιμότητα του στόλου είναι απεριόριστη. Στη συνέχεια, προτείνονται καινοτόμες αντικειμενικές συναρτήσεις, στις οποίες λαμβάνεται υπόψη η παραγωγικότητα των οχημάτων, παρουσιάζοντας έτσι μεγαλύτερο περιθώριο για την εξυπηρέτηση δυναμικών απαιτήσεων που θα παρουσιαστούν στο μέλλον, ειδικά σε περιπτώσεις με σχετικά υψηλή διαθεσιμότητα οχημάτων και μεγάλα χρονικά παράθυρα. Επιπρόσθετα, οι προτεινόμενες μέθοδοι εφαρμόζονται σε πραγματικό σενάριο εταιρείας ταχυμεταφορών και επιδεικνύεται πως αποφέρουν βελτιωμένα αποτελέσματα συγκριτικά με τις χρησιμοποιούμενες πρακτικές δρομολόγησης καθώς και με προηγμένη ευρετική μέθοδο.

Τέλος, μελετάται ενδιαφέρουσα και πρακτική παραλλαγή του ΠΔΔΟΠ που επιτρέπει μεταφόρτωση μεταξύ των οχημάτων κατά τη διάρκεια εκτέλεσης του δρομολογίου, με κύριο στόχο την ανακατανομή του φόρτου εργασίας των «στατικών» παραγγελιών παράδοσης σε πραγματικό χρόνο. Για την επίλυση του προβλήματος αναδρομολόγησης με μεταφόρτωση, προτείνεται καινοτόμο μαθηματικό μοντέλο, καθώς και κατάλληλη ευρετική μέθοδος, ικανή να αντιμετωπίσει περιπτώσεις πρακτικού μεγέθους. Επιπλέον, εκτενής πειραματική διερεύνηση κάτω από διάφορες επιχειρησιακές συνθήκες υποδεικνύει πως η συγκεκριμένη προσέγγιση αποφέρει σημαντικές βελτιώσεις, επιπρόσθετα από αυτές που προσφέρουν οι προηγούμενες προσεγγίσεις.

ΤΟ ΠΡΟΒΛΗΜΑ ΔΥΝΑΜΙΚΗΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΜΕ ΠΑΡΑΛΑΒΕΣ

Το ΠΔΔΟΠ μπορεί να εξηγηθεί πρακτικά θεωρώντας τυπικό σενάριο εταιρείας ταχυμεταφορών, όπως φαίνεται στο Σχήμα Π.1. Συγκεκριμένα, θεωρείται ότι στόλος οχημάτων (σύνολο V) με περιορισμένη χωρητικότητα ανά όχημα $\bar{Q}$ βρίσκεται διαθέσιμος σε κέντρο διανομής. Κατά την αρχή του χρονικού ορίζοντα προγραμματισμού $[0, T_{max}]$, ένα σύνολο οχημάτων $K \subset V$ αναλαμβάνει την εκτέλεση προγραμματισμένων δρομολογίων προκειμένου να εξυπηρετήσει προκαθορισμένο σύνολο (στατικών) πελατών, ενώ το οχημάτων $K_d = V - K$ παραμένει διαθέσιμο στο κέντρο διανομής (Σχ. Π.1α). Κάθε όχημα οφείλει να επιστρέψει στο κέντρο διανομής μέχρι τη χρονική στιγμή $t = T_{max}$. Κατά τη διάρκεια υλοποίησης του πλάνου, νέες απαιτήσεις για παραλαβή (pickup) εισάγονται στο σύστημα (εφεξής ονομάζονται *Δυναμικές Απαιτήσεις, ΔΑ*), οι οποίες θα πρέπει να συλλεχθούν και να επιστραφούν στο κέντρο διανομής για περαιτέρω επεξεργασία (Σχ. Π.2β). Οι *ΔΑ* θα πρέπει να ανατεθούν στα διαθέσιμα οχήματα με άμεση συνέπεια την αναθεώρηση των δρομολογίων τους (Σχ. Π.1γ). Επισημαίνεται πως οι στατικές απαιτήσεις δε μπορούν να ανατεθούν σε άλλο όχημα, ενώ οι ΔΑ μπορούν να εξυπηρετηθούν από κάθε όχημα $V = K \cup K_d$.

Στόχος του ΠΔΔΟΠ είναι η ανάθεση των δυναμικών αυτών απαιτήσεων (παραγγελιών) στα κατάλληλα οχήματα, είτε σε αυτά που ήδη εκτελούν κάποιο δρομολόγιο είτε στα υπόλοιπα που βρίσκονται στο κέντρο διανομής. Σε περίπτωση απουσίας περιορισμού αναφορικά με το διαθέσιμο στόλο, συνήθως ελαχιστοποιείται το κόστος δρομολόγησης, διαφορετικά, σε περίπτωση περιορισμένου στόλου, μεγιστοποιείται ο αριθμός των εξυπηρετούμενων απαιτήσεων. Μία εφικτή λύση του προβλήματος θα πρέπει να εξυπηρετεί όλες τις στατικές απαιτήσεις και να ικανοποιεί τη χωρητικότητα των οχημάτων και του χρονικούς περιορισμούς αναφορικά με τα χρονικά παράθυρα των πελατών και της βάρδιας των οδηγών. Οι δύο διακριτές περιπτώσεις διαθεσιμότητας στόλου (απεριόριστος και περιορισμένος), εξετάζονται ξεχωριστά παρακάτω.



Σχήμα Π.1. Αποτύπωση του ΠΔΔΟΠ; (a) αρχικό πλάνο, (b) άφιξη νέων απαιτήσεων κατά τη διάρκεια εκτέλεσης των δρομολογίων, (c) ενδεικτική λύση μετά την αναδρομολόγηση

Αναδρομολόγηση στο ΠΔΔΟΠ

Η ανάθεση των ΔΑ αντιμετωπίζεται με *περιοδική αναδρομολόγηση* (βλ. Σχ. Π.2). Θεωρείται πως για ολόκληρο τον ορίζοντα $[0, T_{max}]$, θα υλοποιηθούν L *περίοδοι αναδρομολόγησης*, όπου κάθε περίοδος θα αντιστοιχεί σε ένα στατικό πρόβλημα $\Gamma_1, \Gamma_2, \dots, \Gamma_L$, το οποίο θα επιλύεται στις χρονικές στιγμές $T_\ell, \ell = 1,2, \dots, L$ με $T_0 = 0 < T_1 < \cdots < T_L < T_{max} - \tau$. Οι περίοδοι αναδρομολόγησης ($[T_{\ell-1}, T_\ell]$, $\ell \geq 1$) δεν έχουν απαραίτητα την ίδια χρονική διάρκεια. Στο στατικό πρόβλημα που επιλύεται κάθε χρονική στιγμή αναδρομολόγησης T_ℓ, χρησιμοποιείται το σύνολο της πληροφορίας που είναι γνωστή μέχρι εκείνη τη στιγμή. Θεωρείται πως το πρόβλημα (Γ_ℓ) επιλύεται στιγμιαία.

Σχήμα Π.2. Η διαδικασία αναδρομολόγησης

Ένα πρόβλημα αναδρομολόγησης $\Gamma_\ell, \ell \in \{1, \dots, L\}$ λαμβάνει υπόψη του δύο σύνολα απαιτήσεων που δεν έχουν εξυπηρετηθεί ακόμη: i) τις *δεσμευμένες απαιτήσεις* (σύνολο C), που περιλαμβάνουν τις απαιτήσεις που έχουν ανατεθεί σε ένα όχημα και δε μπορούν να μεταβιβαστούν σε άλλα οχήματα, και ii) τις *μη δεσμευμένες απαιτήσεις* (σύνολο F), που αντιστοιχούν σε νέες ΔΑ, ή σε ΔΑ που ελήφθησαν σε προηγούμενες περιόδους και δεν έχουν εξυπηρετηθεί ακόμη, αλλά μπορούν να εξυπηρετηθούν από κάθε όχημα $V = K \cup K_d$. Ανάλογα με τη *τακτική υλοποίησης* (όπως θα συζητηθεί παρακάτω), δύο σενάρια ενδέχεται να ισχύουν: α) οι δεσμευμένες απαιτήσεις αφορούν μόνο σε στατικές απαιτήσεις, ενώ οι ΔΑ που δεν έχουν εξυπηρετηθεί θεωρούνται ως μη δεσμευμένες, και β) οι δεσμευμένες απαιτήσεις περιλαμβάνουν όλες όσες έχουν ανατεθεί σε οχήματα σε προηγούμενες περιόδους και δεν έχουν εξυπηρετηθεί, ενώ ως μη δεσμευμένες ορίζονται μόνο οι νέες ΔΑ.

Η λύση του προβλήματος αναδρομολόγησης στη περίοδο ℓ θεωρεί όλο τον υπολειπόμενο χρονικό ορίζονται $[\mathrm{T}_\ell, \mathrm{T}_{\max}]$. Μέρος της λύσης αυτής υλοποιείται συνεπώς μέχρι την επόμενη χρονική στιγμή αναδρομολόγησης $\mathrm{T}_{\ell+1}$.

Μαθηματικό μοντέλο του προβλήματος αναδρομολόγησης στο ΠΔΔΟΠ

Το παρακάτω μοντέλο περιγράφει το πρόβλημα αναδρομολόγησης αγνοώντας τον δείκτη ℓ, καθότι το πρόβλημα έχει την ίδια μορφή σε κάθε περίοδο αναδρομολόγησης.

Θεωρείται σύνολο $N = C \cup F$ το οποίο αντιπροσωπεύει το σύνολο των απαιτήσεων που δεν έχει εξυπηρετηθεί, με C και F τα σύνολα των δεσμευμένων και μη δεσμευμένων απαιτήσεων, αντίστοιχα, και με $C = \bigcup_{k \in K} C_k$, όπου C_k ορίζεται το σύνολο των δεσμευμένων απαιτήσεων που έχουν ανατεθεί στο όχημα k που εκτελεί ήδη ένα δρομολόγιο (*καθοδόν*). Ορίζεται επίσης σύνολο $M = \bigcup_{k \in K}\{\mu_k\}$, όπου μ_k αντιπροσωπεύει τη παρούσα θέση του οχήματος $k \in K$ και ως 0 ορίζεται ο κόμβος αρχής και τέλους των δρομολογίων (κέντρο διανομής). Επίσης, ο γράφος ορίζεται ως $G = (W, A)$, με $W = C \cup F \cup M \cup \{0\}$ και A το σύνολο των ακμών που συνδέει όλους τους κόμβους $W (A = \{(i,j): i \in W, j \in W \setminus M\})$. Το κόστος διάνυσης μιας ακμής $(i,j), \{i \in W, j \in W \setminus M\}$ ορίζεται ως c_{ij}, ενώ t_{ij} δηλώνεται ο χρόνος διάνυσης της απόστασης μεταξύ δύο κόμβων.

Κάθε απαίτηση $i \in N$ σχετίζεται με τα παρακάτω χαρακτηριστικά:

d_i	Η ζήτηση της απαίτησης του πελάτη (οι επιδόσεις σχετίζονται με αρνητικές τιμές, ενώ οι παραλαβές με θετικές).
s_i	Ο χρόνος εξυπηρέτησης της απαίτησης του πελάτη

h_i	Ο χρόνος άφιξης της απαίτησης i. Προφανώς, $0 < h_i < T_{max} - \tau, \forall i \in F$ και $h_i = 0, \forall i \in C$
$[a_i, b_i]$	Το χρονικό παράθυρο της απαίτησης. Για τις στατικές απαιτήσεις, $0 \leq a_i < b_i \leq T_{max}$ ενώ για τις ΔΑ, $h_i < a_i < b_i \leq T_{max}$.

Το μοντέλο περιλαμβάνει τρία διαφορετικά σύνολα μεταβλητών: i) τη μεταβλητή x_{ijk} που ισούται με 1 αν το όχημα $k \in V$ διανύει την ακμή $(i,j) \in A$ και μηδέν σε άλλη περίπτωση, ii) τη μεταβλητή w_{ik}, που αντιπροσωπεύει το χρόνο έναρξης εξυπηρέτησης της απαίτησης (κόμβου) $i \in N$ από το όχημα $k \in V$, ενώ για το κέντρο διανομής, $w_{0k} \geq T$, και iii) τη μεταβλητή Q_{ik}, που δηλώνει το φορτίο του οχήματος $k \in V$ αμέσως μετά την εξυπηρέτηση του κόμβου $i \in W$.

Περίπτωση απεριόριστου στόλου οχημάτων

Αντικειμενικός στόχος του προβλήματος στην περίπτωση απεριόριστου πλήθους διαθέσιμων οχημάτων είναι η ελαχιστοποίηση του συνολικού κόστους δρομολόγησης καθ' όλο το εύρος του χρονικού ορίζοντα $[\mathrm{T}_\ell, \mathrm{T}_{\max}]$ και δίνεται από τη συνάρτηση (Π.1) παρακάτω:

$$\min(z) = \sum_{k \in V} \sum_{(i,j) \in A} c_{ijk} x_{ijk} \qquad (\Pi.1)$$

Υπό τους περιορισμούς:

$$\sum_{j \in C_k \cup F \cup \{0\}} x_{ijk} = 1 \qquad \forall k \in K, \forall i \in C_k \cup \{\mu_k\} \qquad (\Pi.2)$$

$$\sum_{k \in V} \sum_{j \in W} x_{ijk} = 1 \qquad \forall i \in F \qquad (\Pi.3)$$

$$\sum_{i \in C_k \cup F \cup \{\mu_k\}} x_{i0k} = 1 \qquad \forall k \in K \qquad (\Pi.4)$$

$$\sum_{j \in F} x_{0jk} \leq 1 \qquad \forall k \in K^d \qquad (\Pi.5)$$

$$\sum_{j \in F} x_{0jk} = \sum_{j \in F} x_{j0k} \qquad \forall k \in K^d \qquad (\Pi.6)$$

$$\sum_{i \in W} x_{ihk} - \sum_{j \in W} x_{hjk} = 0 \qquad \forall h \in N, \forall k \in V \qquad (\Pi.7)$$

$$Q_{jk} \geq Q_{ik} + d_j - Z(1 - x_{ijk}) \qquad \forall (i,j) \in A, \forall k \in V \qquad (\Pi.8)$$

$$\max\{0, d_i\} \leq Q_{ik} \leq \min\{\bar{Q}, \bar{Q} + d_i\} \qquad \forall i \in N, \forall k \in V \qquad (\Pi.9)$$

$$\mathrm{w_{jk}} \geq \mathrm{w_{ik}} + \mathrm{s_i} + \mathrm{t_{ij}} - \mathrm{Z}(1 - \mathrm{x_{ijk}}) \qquad \forall (i,j) \in A, \forall k \in V \qquad (\Pi.10)$$

$$\max(\mathrm{a_i}, \mathrm{T}) \sum_{\mathrm{j \in W}} \mathrm{x_{ijk}} \leq \mathrm{w_{ik}} \leq \mathrm{b_i} \sum_{\mathrm{j \in W}} \mathrm{x_{ijk}} \qquad \forall k \in V, \forall i \in W \qquad (\Pi.11)$$

$$\mathrm{T} \leq \mathrm{w_{0k}} \leq \mathrm{b_0} \qquad \forall k \in K^d \qquad (\Pi.12)$$

$$\mathrm{x_{ijk}} \in \{0,1\} \qquad \forall (i,j) \in A, \forall k \in V \qquad (\Pi.13)$$

Η αντικειμενική συνάρτηση (Π.1) αφορά στο συνολικό κόστος δρομολόγησης. Ο περιορισμός (Π.2) εξασφαλίζει ότι κάθε όχημα k (καθοδόν) θα πρέπει να εξυπηρετήσει όλες τις δεσμευμένες απαιτήσεις που του έχουν ανατεθεί. Ο περιορισμός (Π.3) ορίζει πως όλες οι μη δεσμευμένες απαιτήσεις θα εξυπηρετηθούν, είτε από ένα όχημα καθοδόν, είτε από όχημα που βρίσκεται στο κέντρο διανομής. Ο περιορισμός (Π.4) ορίζει πως κάθε καθοδόν όχημα θα πρέπει να επιστρέψει στο κέντρο διανομής. Ο περιορισμός (Π.5) ορίζει πως είναι δυνατό νέα οχήματα να αποσταλούν από το κέντρο διανομής κατά την αναδρομολόγηση για να καλύψουν ΔΑ, ενώ ο περιορισμός (Π.6) αναγκάζει τα οχήματα αυτά να επιστρέψουν στο κέντρο διανομής. Οι περιορισμοί (Π.7) αναφέρονται στη διατήρηση ροής κάθε οχήματος. Μέσω των περιορισμών (Π.8) και (Π.9) καθορίζεται πως το φορτίο κάθε οχήματος δε θα υπερβεί την χωρητικότητά του (όπου Z ένας μεγάλος θετικός αριθμός). Οι περιορισμοί (Π.10) και (Π.11) καθορίζουν πως κάθε απαίτηση εξυπηρετείται εντός του χρονικού παραθύρου της, ενώ οι περιορισμοί (Π.12) εξασφαλίζουν πως νέα οχήματα, που δυνητικά θα αποσταλούν από το κέντρο διανομής, θα εκκινήσουν μετά από τη χρονική στιγμή αναδρομολόγησης και θα επιστρέψουν μέσα στον επιτρεπτό χρονικό ορίζοντα. Τέλος, οι περιορισμοί (Π.13) δεσμεύουν τις μεταβλητές ροής σε δυαδικές τιμές $\{0, 1\}$.

<u>Περίπτωση περιορισμένου στόλου οχημάτων</u>

Στη περίπτωση όπου το πλήθος οχημάτων του στόλου είναι περιορισμένο, είναι πιθανό να μην εξυπηρετηθούν όλες οι ΔΑ. Συνεπώς, ορισμένες τροποποιήσεις απαιτούνται στο προαναφερόμενο (γενικό) μοντέλο. Η πρώτη τροποποίηση αφορά το περιορισμό αναφορικά με την εξυπηρέτηση των ΔΑ. Συγκεκριμένα, ο περιορισμός (Π.3) μπορεί να μετατραπεί στον (Π.14) παρακάτω:

$$\sum_{k \in V} \sum_{j \in W} x_{ijk} \leq 1 \qquad \forall i \in F \qquad (\Pi.14)$$

Η δεύτερη τροποποίηση αφορά την αντικειμενική συνάρτηση (Π.1). Η ελαχιστοποίηση του κόστους δεν είναι πλέον κατάλληλος αντικειμενικός στόχος, εφόσον σε συνδυασμό με τον (Π.14), δε θα εξυπηρετούσε καμία ΔΑ. Ένα καταλληλότερος αντικειμενικός στόχος θα μπορούσε να είναι η μεγιστοποίηση εξυπηρέτησης των ΔΑ, σε συνδυασμό με το ελάχιστο κόστος, όπως φαίνεται στην (Π.15) παρακάτω:

$$\min(z) = -\xi_u \sum_{k \in V} \sum_{(i,j) \in A | i \in F, j \in W} x_{ijk} + \sum_{k \in V} \sum_{(i,j) \in A} c_{ijk} x_{ijk} \quad (Π.15)$$

όπου ξ_u υποδηλώνει ένα «κέρδος» για κάθε ΔΑ που εξυπηρετείται. Η καταλληλόλητα της παρούσας αντικειμενικής, καθώς και η περίπτωση του περιορισμένου στόλου, εξετάζονται αναλυτικότερα στην σχετική ενότητα παρακάτω.

Τέλος, προστίθεται ο παρακάτω περιορισμός (Π.16) που αφορά το πλήθος των διαθέσιμων οχημάτων που χρησιμοποιούνται κάθε δεδομένη στιγμή.

$$\sum_{k \in V} \sum_{i \in W} x_{i0k} \leq |V| \quad (Π.16)$$

ΔΙΑΣΠΑΣΗ ΤΟΥ ΜΑΘΗΜΑΤΙΚΟΥ ΜΟΝΤΕΛΟΥ ΑΝΑΔΡΟΜΟΛΟΓΗΣΗΣ

Για την βέλτιστη επίλυση του παραπάνω μοντέλου μεικτού γραμμικού προγραμματισμού, αρχικά επιλύεται η γραμμική «χαλάρωση» του ανωτέρω προβλήματος μέσω της μεθόδου ΔΔΜ για την εύρεση κατώτατων ορίων (lower bounds). Η ΔΔΜ διασπά το χαλαρωμένο μοντέλο σε ένα *Κυρίως Πρόβλημα* (ΚΠ) και πολλαπλά *Υπό-Προβλήματα* (ΥΠ). Για την εύρεση ακέραιων λύσεων χρησιμοποιείται η μέθοδος branch-and-price, στην οποία η ΔΔΜ χρησιμοποιείται σε κάθε κόμβο του σχετικού δέντρου.

Στη παρούσα ενότητα παρουσιάζεται η διάσπαση του μαθηματικού μοντέλου για τη γενική περίπτωση του ΠΔΔΟΠ (απεριόριστος στόλος). Στη περίπτωση αυτή, το ΚΠ περιλαμβάνει μόνο τους περιορισμούς αναφορικά με την εξυπηρέτηση των απαιτήσεων. Τα ΥΠ περιλαμβάνουν τους λοιπούς περιορισμούς αναφορικά με την εφικτότητα των δρομολογίων. Οι τροποποιήσεις του μοντέλου για την αντιμετώπιση του περιορισμού αναφορικά με το στόλο οχημάτων περιγράφονται σε επόμενη ενότητα.

Το προτεινόμενο Κυρίως Πρόβλημα (ΚΠ)

Το ΚΠ για το ΠΔΔΟΠ μοντελοποιείται συνήθως ως ένα πρόβλημα διαμερισμού συνόλου (set partitioning problem, SPP), του οποίου κάθε μεταβλητή (κολώνα) αντιστοιχεί σε ένα *εφικτό* δρομολόγιο και κάθε περιορισμός αντιστοιχεί σε μία απαίτηση που εξυπηρετείται. Συνεπώς, ορίζεται δυαδική μεταβλητή a_{ir} η οποία ισούται με 1 αν η απαίτηση $i \in N$ εξυπηρετείται από το δρομολόγιο $r \in \Omega$ και μηδέν σε κάθε άλλη περίπτωση, καθώς επίσης και συντελεστές y_r οι οποίοι ισούνται με 1 αν το δρομολόγιο $r \in \Omega$ χρησιμοποιείται από τη λύση και μηδέν διαφορετικά. Θεωρώντας πως το c_r δηλώνει το κόστος του δρομολογίου $r \in \Omega$, τότε η αντικειμενική του ΚΠ έχει την ακόλουθη μορφή:

Ελαχιστοποίηση $$\sum_{r \in \Omega} c_r y_r \qquad (\Pi.17)$$

Υπό τους περιορισμούς: $$\sum_{r \in \Omega} a_{ir} y_r = 1 \qquad \forall i \in N \qquad (\Pi.18)$$

$$y_r = \{0, 1\} \qquad \forall r \in \Omega \qquad (\Pi.19)$$

Συνεπώς, το ΚΠ περιλαμβάνει μόνο εκείνους τους περιορισμούς που επιβάλλουν μοναδική εξυπηρέτηση σε κάθε απαίτηση. Οι υπόλοιποι περιορισμοί αντιμετωπίζονται από τα ΥΠ. Στη παρούσα μοντελοποίηση, το σύνολο Ω όλων των εφικτών δρομολογίων (μεταβλητών), αποτελείται από δύο υποσύνολα, $\Omega = (\bigcup_{k \in K} \Omega_k) \cup \Omega_p$, όπου: α) Ω_k αφορά το υποσύνολο των δρομολογίων που λαμβάνουν χώρα από τα καθοδόν οχήματα K (κάθε τέτοιο δρομολόγιο εκκινεί από τη παρούσα θέση του οχήματος, καταλήγει στο κέντρο διανομής και περιλαμβάνει όλες τις δεσμευμένες C_k απαιτήσεις και, πιθανώς, ορισμένες μη δεσμευμένες $F' \subseteq F$), και β) Ω_p που αφορά στο σύνολο των δρομολογίων για τα οχήματα K_d που βρίσκονται στο κέντρο διανομής (τα δρομολόγια αυτά εκκινούν και καταλήγουν στο κέντρο διανομής και συμπεριλαμβάνουν μόνο F απαιτήσεις).

Λόγω του πλήθους των δυνατών συνδυασμών απαιτήσεων, ορίζουμε ως Ω' ένα υποσύνολο του Ω που περιλαμβάνει γνωστά και εφικτά δρομολόγια (Περιορισμένο Κυρίως Πρόβλημα, ΠΚΠ). Για τη κατασκευή αυτού του υποσυνόλου, χρησιμοποιείται η πληροφορία από τη λύση της προηγούμενης περιόδου αναδρομολόγησης (που περιλαμβάνει εφικτά δρομολόγια), αφαιρώντας τις απαιτήσεις που έχουν ήδη εξυπηρετηθεί στο διάστημα $[T_{\ell-1}, T_\ell]$. Για τις νέες ΔΑ που έχουν αφιχθεί στο διάστημα $[T_{\ell-1}, T_\ell]$, δημιουργείται ένα δρομολόγιο ανά ΔΑ ($[depot - i - depot], \forall i \in F$) και προστίθεται στο υποσύνολο Ω'. Συνεπώς, με βάση αυτό το υποσύνολο εφικτών δρομολογίων επιλύεται γραμμική χαλάρωση (θεωρώντας δεκαδικές τιμές για τις μεταβλητές y_r αντί για δυαδικές) του προβλήματος στη παρούσα περίοδο αναδρομολόγησης.

Τα Υποπροβλήματα

Προκειμένου να αναγνωριστούν νέες μεταβλητές (δρομολόγια) με αρνητικό μειωμένο κόστος αναφορικά με τη λύση του ΠΚΠ, επιλύεται διαφορετικό πρόβλημα βελτιστοποίησης (ΥΠ). Στο πρόβλημα αυτό λαμβάνονται υπόψη όλοι οι περιορισμοί εφικτότητας ενός δρομολογίου (όπως για παράδειγμα, η απαίτηση εξυπηρέτησης όλων των δεσμευμένων απαιτήσεων που έχουν ανατεθεί σε ένα όχημα, οι χρονικοί περιορισμοί, καθώς και οι περιορισμοί φορτίου). Στη παρούσα διατριβή, προτείνεται μέθοδος ακριβούς επίλυσης (exact) του ΥΠ, καθώς και ευρετική μέθοδος (heuristic).

Για τη μέθοδο ακριβούς επίλυσης, το ΥΠ μοντελοποιείται ως ένα Στοιχειώδες Πρόβλημα Συντομότερης Διαδρομής με Χρονικά Παράθυρα και Περιορισμούς Χωρητικότητας (ΣΠΣΔΧΠΠΧ) και επιλύεται με μεθόδους Δυναμικού Προγραμματισμού. Για τη περίπτωση της ευρετικής μεθόδου, χρησιμοποιείται κατάλληλη μέθοδος παρεμβολής (insertion-based heuristic) η οποία εκμεταλλεύεται τη πληροφορία από τις δυικές τιμές που προκύπτουν από την επίλυση της γραμμικής χαλάρωσης του ΠΚΠ. Ανεξαρτήτως της μεθόδου, η λύση του ΥΠ καταλήγει με ένα ή περισσότερα δρομολόγια (κολώνες) τα οποία ελαχιστοποιούν μία δεδομένη αντικειμενική συνάρτηση. Τα δρομολόγια που προκύπτουν από την επίλυση του ΥΠ ενσωματώνονται σε αυτά του ΠΚΠ προκειμένου να επιλυθεί ξανά. Η διαδικασία αυτή επαναλαμβάνεται έως ότου η λύση του ΠΚΠ είναι μη αρνητική.

<u>Μέθοδος ακριβούς επίλυσης των ΥΠ</u>

Για το ΠΔΔΟΠ, ορίζουμε και επιλύουμε $|K|+1$ ανεξάρτητα ΥΠ, ένα για κάθε καθοδόν όχημα K (δηλαδή για τη δημιουργία των δρομολογίων Ω_k), καθώς και ξεχωριστό ΥΠ που αντιστοιχεί σε όλα τα οχήματα που βρίσκονται στο κέντρο διανομής (δηλαδή για τη δημιουργία των δρομολογίων Ω_p). Κάθε ΥΠ μοντελοποιείται ως ένα ΣΠΣΔΧΠΠΧ και επιλύεται με τον *αλγόριθμο διόρθωσης ετικετών* (*label correcting algorithm*). Κάθε ένα από τα $k=1,2,\dots,|K|$ ΥΠ θεωρούν απαιτήσεις $N_k=C_k\cup F$ (όπου C_k αντιπροσωπεύει τις δεσμευμένες απαιτήσεις που έχουν ανατεθεί στο όχημα $k\in K$), ενώ το $|K|+1$ ΥΠ θεωρεί απαιτήσεις $N_{|K|+1}=F$. Η ερευνητική συνεισφορά της παρούσας μεθόδου έγκειται στην ενίσχυση των κριτηρίων κυριαρχίας, με αποτέλεσμα την απόρριψη ετικετών σε πρώιμο στάδιο και την αποτελεσματικότερη εύρεση της βέλτιστης λύσης.

<u>Ευρετική μέθοδος επίλυσης των ΥΠ</u>

Στην προτεινόμενη νέα ευρετική μέθοδο, θεωρείται η λύση της προηγούμενης περιόδου αναδρομολόγησης και δημιουργούνται νέα δρομολόγια (κολώνες) για τα καθοδόν οχήματα ενσωματώνοντας της απαιτήσεις F στα υφιστάμενα δρομολόγια μέσω αλγορίθμου παρεμβολής (insertion) ο οποίος βασίζεται στις δυικές τιμές που προκύπτουν από την επίλυση του εκάστοτε ΠΚΠ. Για τη δημιουργία νέων δρομολογίων για τα οχήματα που βρίσκονται στο κέντρο διανομής, επιλύεται ένα ΣΠΣΔΧΠΠΧ χρησιμοποιώντας περιορισμένη (ευρετική) παραλλαγή του αλγορίθμου διόρθωσης ετικετών.

Συνδυάζοντας το Περιορισμένο Κυρίως Πρόβλημα με τα Υποπροβλήματα

Όπως προαναφέρθηκε, αν με την επίλυση των ΥΠ, δημιουργηθεί έστω και ένα δρομολόγιο με αρνητικό μειωμένο κόστος (είτε για καθοδόν όχημα, είτε για όχημα από το κέντρο διανομής), τότε το δρομολόγιο προστίθεται στο ΠΚΠ και η γραμμική χαλάρωση του νέου ΠΚΠ επιλύεται ξανά. Αν σε κάποια επανάληψη της διαδικασίας αυτής δε βρεθούν νέα δρομολόγια με αρνητικό μειωμένο κόστος, τότε η διαδικασία ολοκληρώνεται και το βέλτιστο κατώτατο όριο (lower bound) έχει επιτευχθεί. Επισημαίνεται και πάλι πως η προαναφερόμενη διαδικασία ΔΔΜ επιλύει τη γραμμική χαλάρωση του ΠΚΠ. Για την επίτευξη ακέραιων λύσεων, η ΔΔΜ ενσωματώνεται σε πλαίσιο Branch & Bound. Εν γένει, η ενσωμάτωση της ΔΔΜ με Branch & Bound, συνιστά τον αλγόριθμο Branch-and-Price.

ΣΤΡΑΤΗΓΙΚΕΣ ΑΝΑΔΡΟΜΟΛΟΓΗΣΗΣ (ΑΠΕΡΙΟΡΙΣΤΟΣ ΣΤΟΛΟΣ)

Εισάγεται η έννοια της *στρατηγικής αναδρομολόγησης*, η οποία αποτελείται από τον συνδυασμό α) της *πολιτικής αναδρομολόγησης* (re-optimization policy) που σχετίζεται με τη συχνότητα αναδρομολόγησης και, β) της *τακτικής υλοποίησης* (implementation tactic), που αφορά στο τμήμα του νέου δρομολογίου που κοινοποιείται στο στόλο προς υλοποίηση.

Διερευνώνται διαφορετικές πολιτικές αναδρομολόγησης αναφορικά με τον αριθμό των ΔΑ που έχουν αφιχθεί μεταξύ δύο διαδοχικών περιόδων αναδρομολόγησης:

- *Αναδρομολόγηση ανά κάθε απαίτηση* (Single-request re-optimization, SRR)*:* Άμεση αναδρομολόγηση με την άφιξη κάθε ΔΑ
- *Αναδρομολόγηση ανά αριθμό απαιτήσεων* (N-request re-optimization, NRR)*:* Αναδρομολόγηση μετά την άφιξη ενός προκαθορισμένου αριθμού N (N>1) ΔΑ
- *Αναδρομολόγηση καθορισμένου χρόνου* (Fixed-Time Re-optimization, FTR): Αναδρομολόγηση σε προκαθορισμένες χρονικές στιγμές (π.χ. κάθε μία ώρα).

Επιπρόσθετα, διερευνώνται δύο βασικές τακτικές υλοποίησης του νέου πλάνου:

- *Τακτική πλήρους κοινοποίησης* (Full-Release tactic, FR): Όλες οι ΔΑ μετά την αναδρομολόγηση κοινοποιούνται στο στόλο άμεσα και δε μπορούν να αναδρομολογηθούν σε μελλοντικές περιόδους.
- *Τακτική μερικής κοινοποίηση* (Partial-Release tactic, FR): Μόνο οι ΔΑ που έχουν προγραμματιστεί προς υλοποίηση έως την επόμενη περίοδο αναδρομολόγησης κοινοποιούνται (στην πράξη σταδιακά μία προς μία έως την επόμενη περίοδο αναδρομολόγησης). Οι υπόλοιπες ΔΑ θεωρούνται προς αναδρομολόγηση σε μελλοντικές

περιόδους. Πρακτικά, αυτό σημαίνει πως οι ΔΑ που δεν έχουν εξυπηρετηθεί μέχρι τη χρονική στιγμή αναδρομολόγησης θεωρούνται εκ νέου ως μη δεσμευμένες και ανήκουν στο σύνολο F.

Η παρούσα διατριβή εξετάζει επίσης θεωρητικά ζητήματα αναφορικά με την αναμενόμενη συμπεριφορά των στρατηγικών αναδρομολόγησης. Συγκεκριμένα αποδεικνύεται ότι: α) στη περίπτωση ενός οχήματος, και οι δύο τακτικές υλοποίησης αναμένεται να αποφέρουν τα ίδια αποτελέσματα, β) το κόστος δρομολόγησης που αντιστοιχεί την τακτική PR είναι πάντα χαμηλότερο (ή ίσο) από εκείνο που αντιστοιχεί στην τακτική FR για τις πρώτες δύο περιόδους αναδρομολόγησης ($\ell < 3$), γ) για $\ell \geq 3$ και ειδικά αν ένα ή περισσότερα οχήματα έχουν αποσταλεί από το κέντρο διανομής σε οποιαδήποτε περίοδο $\ell > 0$, δεν είναι βέβαιο ότι το κόστος της τακτικής PR τακτική είναι χαμηλότερο από εκείνο της τακτικής FR.

Πειραματική διερεύνηση

Το κριτήριο μέτρησης ποιότητας της λύσης

Για τη μέτρηση ποιότητας της λύσης, χρησιμοποιείται η μετρική Value of Information (VoI, Mitrovic-Minic *et al.*, 2004), η οποία ισούται με την ποσοστιαία διαφορά του δυναμικού προβλήματος από τη θεωρητική λύση του στατικού προβλήματος. Το τελευταίο αντιστοιχεί στην περίπτωση κατά την οποία το σύνολο των ΔΑ είναι γνωστές πριν την εκκίνηση των οχημάτων από το κέντρο διανομής (στον χρόνο $t = 0$).

Πειραματικές περιπτώσεις που χρησιμοποιήθηκαν

Για τη πειραματική διερεύνηση του ΠΔΔΟΠ, χρησιμοποιήθηκαν τα σύνολα προβλημάτων R1, C1 και RC1 του Solomon (1987). Επίσης, χρησιμοποιήθηκαν τα σύνολα προβλημάτων MR2, MC2 και MRC2 των Kontoravdis and Bard (1995), που χρησιμοποιούν τα χαρακτηριστικά των προβλήματα R2, C2 και RC2 του Solomon, αλλά με μειωμένη χωρητικότητα οχημάτων. Συμπεριλαμβάνονται επίσης και τα πειράματα vrpnc8 και vrpnc14 των Christofides *et al.* (1979) τα οποία δεν έχουν χρονικά παράθυρα, αλλά χρησιμοποιούν ίδιες συντεταγμένες πελατών όπως τα σύνολα R1 και C1 (τα πειράματα αυτά αναφέρονται εφεξής ως R100 και C100).

Με βάση τη παραπάνω σειρά πειραμάτων, μπορούν να αναλυθούν παράμετροι όπως α) η *γεωγραφική κατανομή πελατών*, και β) τα *χρονικά παράθυρα*. Επιπρόσθετα εξετάζεται το δυναμικό περιεχόμενο των προβλημάτων (δηλαδή, το ποσοστό των ΔΑ στο σύνολο όλων των απαιτήσεων του πειράματος). Για το λόγο αυτό, εξετάζονται τρεις τιμές δυναμικού περιεχομένου 25%, 50% και 75% για τα για τα σύνολα πειραμάτων R1, C1 και RC1, καθώς και 50% για τα MR2, MC2 και MRC2. Συνεπώς, εξετάζονται συνολικά 120 περιπτώσεις προβλημάτων (3 τιμές δυναμικού περιεχομένου για τα 31 πειράματα των R1, C1, RC1 και μία τιμή για τα 27 πειράματα των MR2, MC2 και MRC2). Για κάθε μία από τις 120 περιπτώσεις προβλημάτων, δημιουργούνται 10 διαφορετικά προβλήματα (διαφορετική επιλογή απαιτήσεων ως στατικών και ΔΑ), με αποτέλεσμα τη δημιουργία 1,200 προβλημάτων συνολικά.

Αξιολόγηση της ευρετικής μεθόδου επίλυσης αναφορικά με τη βέλτιστη λύση

Αρχικά, εξετάζεται η απόδοση της ευρετικής μεθόδου επίλυσης σε σχέση με τη βέλτιστη λύση για μεγάλο μέρος των προαναφερόμενων προβλημάτων. Για κάθε πρόβλημα, θεωρήθηκε πως α) όλες οι στατικές απαιτήσεις έχουν ανατεθεί στα οχήματα (και δε μπορούν να ανατεθούν σε άλλο όχημα πέρα από το αρχικό), και β) όλες οι ΔΑ είναι γνωστές προ της εκκίνησης των οχημάτων από το κέντρο διανομής.

Ο Πίνακας Π.1 παρουσιάζει τα αποτελέσματα ανά σύνολο προβλημάτων, ως το μέσο όρο όλων των πειραμάτων και προβλημάτων ανά σύνολο. Ο Πίνακας παρουσιάζει τη ποσοστιαία απόκλιση της ευρετικής λύσης (HEUR) από τη βέλτιστη (OPT) για κάθε μία από τις τιμές δυναμικού περιεχομένου 25% και 50%. Παρουσιάζονται επίσης οι αντίστοιχοι υπολογιστικοί χρόνοι των δύο μεθόδων ανά τιμή (σε δευτερόλεπτα). Η τελευταία στήλη δίνει την απόκλιση της μεθόδου HEUR ανά σύνολο προβλημάτων κατά μέσο όρο για τις δύο τιμές δυναμικού περιεχομένου. Σύμφωνα με τον Πίνακα, η μέθοδος HEUR φαίνεται να παρέχει ανταγωνιστικές λύσεις, με μέση απόκλιση 2.2% από τη βέλτιστη. Αναφορικά με τους υπολογιστικούς χρόνους, η μέθοδος φαίνεται να είναι ιδιαίτερα αποτελεσματική, συγκριτικά με τη μέθοδο ακριβούς επίλυσης.

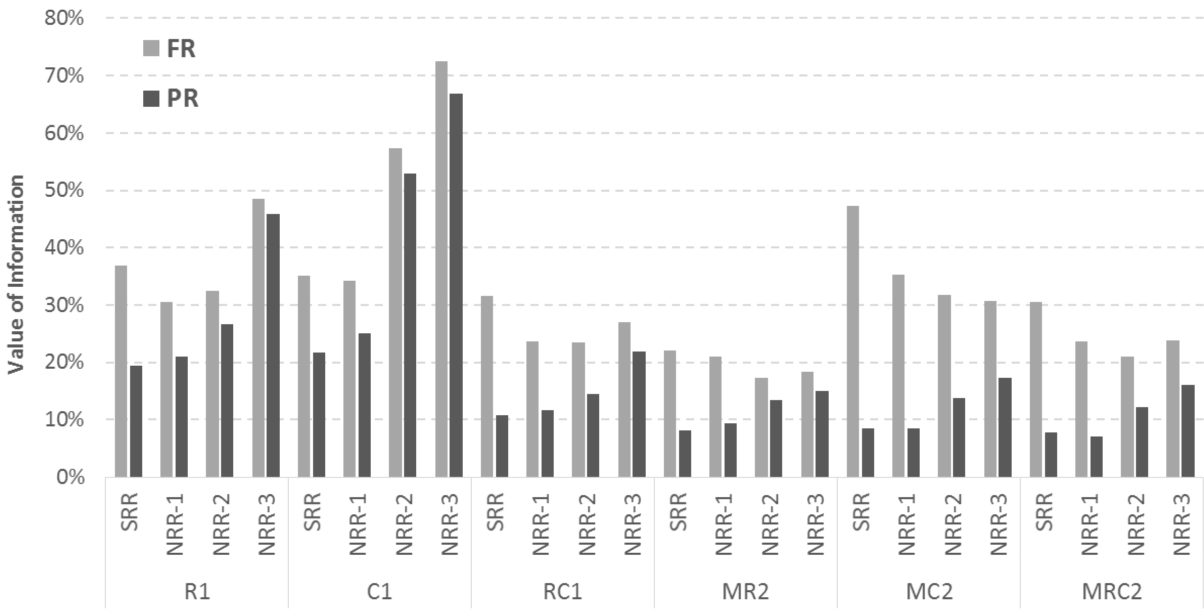
Πίνακας Π.1. Απόδοση της ευρετικής μεθόδου επίλυσης

Dataset	Nodes	$dod = 25\%$			$dod = 50\%$			Average %Dev
		%Dev	CT_{OPT}	CT_{HEUR}	%Dev	CT_{OPT}	CT_{HEUR}	
R1	100	2.0%	719.3	36.8	1.8%	5239.5	56.6	*1.9%*
C1	100	2.6%	136.1	24.8	2.5%	2029.0	68.6	*2.6%*
RC1	100	2.5%	188.4	32.3	2.0%	896.1	35.7	*2.3%*
MR2	50	2.1%	651.0	13.1	2.1%	6108.1	94.9	*2.1%*
MC2	50	1.4%	632.9	10.6	1.9%	3509.9	140.6	*1.7%*
MRC2	50	2.7%	382.3	8.7	2.2%	1031.3	75.5	*2.5%*
Average		2.2%	451.7	21.1	2.1%	3135.7	78.7	2.2%

Πειραματική διερεύνηση των στρατηγικών αναδρομολόγησης

Η παρούσα πειραματική διερεύνηση περιλαμβάνει όλα τα πειράματα που περιεγράφηκαν παραπάνω. Αφορά αρχικά την συνολική συμπεριφορά των προτεινόμενων στρατηγικών αναδρομολόγησης και στη συνέχεια τη συμπεριφορά σε σχέση με διάφορες βασικές παραμέτρους του συστήματος. Για τη διερεύνηση αυτή χρησιμοποιήθηκαν οι πολιτικές SRR και NRR. Αναφορικά με τις πολιτικές NRR, χρησιμοποιήθηκαν οι NRR-1, NRR-2 και NRR-3, που αναφέρονται σε αναδρομολόγηση κάθε 10%, 20% και 33% των ΔΑ που εισάγονται στο σύστημα, αντίστοιχα. Κάθε πολιτική εξετάστηκε κάτω από τις τακτικές FR και PR (άρα, συνολικά εξετάζονται οχτώ στρατηγικές αναδρομολόγησης).

Στο Σχήμα Π.3 παρουσιάζεται η απόδοση (αναφορικά με το VoI) των στρατηγικών για κάθε σύνολο προβλημάτων, ως μέσος όρος όλων των προβλημάτων και τιμών δυναμικού περιεχομένου. Για ομοιογένεια των αποτελεσμάτων, όλα τα πειράματα επιλύθηκαν με την ευρετική μέθοδο (HEUR). Από το Σχήμα είναι προφανές πως: α) η στρατηγική SRR-PR παρέχει τα καλύτερα αποτελέσματα κατά μέσο όρο (το ελάχιστο VoI), και β) η τακτική PR παρέχει καλύτερα αποτελέσματα κατά μέσο όρο από την FR για όλα τα σύνολα προβλημάτων. Η διαφορά των δύο τακτικών τείνει να μειώνεται με την αύξηση της συχνότητας αναδρομολόγησης.



Σχήμα Π.3. Μέση απόδοση των στρατηγικών αναδρομολόγησης για κάθε σύνολο προβλημάτων

Επιπρόσθετα, η διατριβή επικεντρώνεται στη συμπεριφορά των στρατηγικών αναφορικά τρεις βασικές παραμέτρους: α) τα χρονικά παράθυρα πελατών, β) το δείκτη δυναμικού περιεχομένου, και γ) το περιθώριο απόκρισης από την άφιξη της ΔΑ μέχρι το επιτρεπτό όριο εξυπηρέτησής της. Η εν λόγω διερεύνηση παρείχε τα εξής (γενικά) συμπεράσματα:

α) Όταν επιτρέπεται από το εκάστοτε επιχειρησιακό σενάριο, θα πρέπει να προτιμάται η τακτική PR με όσο το δυνατό μεγαλύτερη συχνότητα αναδρομολόγησης.

β) Όταν η τακτική FR είναι αναπόφευκτη, λόγω των χαρακτηριστικών του περιβάλλοντος, η αναδρομολόγηση θα πρέπει να λαμβάνει χώρα σε i) μικρά έως μέτρια χρονικά διαστήματα για περιπτώσεις μικρού εύρους χρονικών παραθύρων ή μικρού περιθωρίου απόκρισης και ii) μέτρια έως μεγάλα διαστήματα για περιπτώσεις μεγάλους εύρους παραθύρων ή περιθωρίου απόκρισης.

γ) Σε προβλήματα με υψηλό δυναμικό περιεχόμενο, συνιστάται μέτρια συχνότητα αναδρομολόγησης (ανεξαρτήτως τακτικής υλοποίησης).

Τέλος, διερευνήθηκε κατά πόσο η χρήση βέλτιστης επίλυσης σε κάθε περίοδο αναδρομολόγησης αποφέρει καλύτερα αποτελέσματα σε ολόκληρο το δυναμικό πρόβλημα (πολλαπλές περίοδοι). Τα αποτελέσματα υποδεικνύουν πως η βέλτιστη επίλυση του προβλήματος σε κάθε περίοδο ενδέχεται να οδηγήσει σε δυσμενέστερα αποτελέσματα από εκείνα που αντιστοιχούν στην πρακτική επίλυσης του προβλήματος αναδρομολόγησης μέσω της ευρετικής μεθόδου (για κάθε περίοδο αναδρομολόγησης), ιδιαίτερα σε περιπτώσεις με διευρυμένο πεδίο λύσεων (π.χ. μεγάλο εύρος χρονικών παραθύρων).

ΤΟ ΠΔΔΟΠ ΓΙΑ ΤΗ ΠΕΡΙΠΤΩΣΗ ΠΕΡΙΟΡΙΣΜΕΝΟΥ ΣΤΟΛΟΥ ΟΧΗΜΑΤΩΝ

Η περίπτωση κατά την οποία ο στόλος οχημάτων είναι περιορισμένος και, συνεπώς, κάποιες από τις ΔΑ ενδέχεται να μη μπορούν να εξυπηρετηθούν, ορίζεται ως το *ΠΔΔΟΠ με Περιορισμούς Πόρων* (ΠΔΔΟΠΠΠ). Για την αντιμετώπιση του ΠΔΔΟΠΠΠ, επεκτείνεται η προτεινόμενη B&P μέθοδος για την επίλυση του σχετικού προβλήματος αναδρομολόγησης. Εξετάζονται εναλλακτικές αντικειμενικές συναρτήσεις που προσπαθούν να μεγιστοποιήσουν την εξυπηρέτηση των απαιτήσεων, ενώ ταυτόχρονα μεγιστοποιούν τη παραγωγικότητα των οχημάτων. Τόσο η αρχική μοντελοποίηση του ΠΔΔΟΠ, όσο και η διαδικασία επίλυσης τροποποιούνται ανάλογα ώστε να είναι σε θέση να αντιμετωπίσουν τον περιορισμό του πλήθους οχημάτων. Η απόδοση των προτεινόμενων αντικειμενικών συναρτήσεων εξετάζεται συγκριτικά με την συμβατική αντικειμενική συνάρτηση που στοχεύει βασικά στη μεγιστοποίηση των εξυπηρετούμενων απαιτήσεων (βλ. Π.15), εξετάζοντας πλήθος επιχειρησιακών σεναρίων και παραμέτρων. Η προτεινόμενη μέθοδος επίλυσης εφαρμόζεται επίσης σε πρακτικό περιβάλλον μεγάλης εταιρείας ταχυμεταφορών.

Προτεινόμενες αντικειμενικές συναρτήσεις

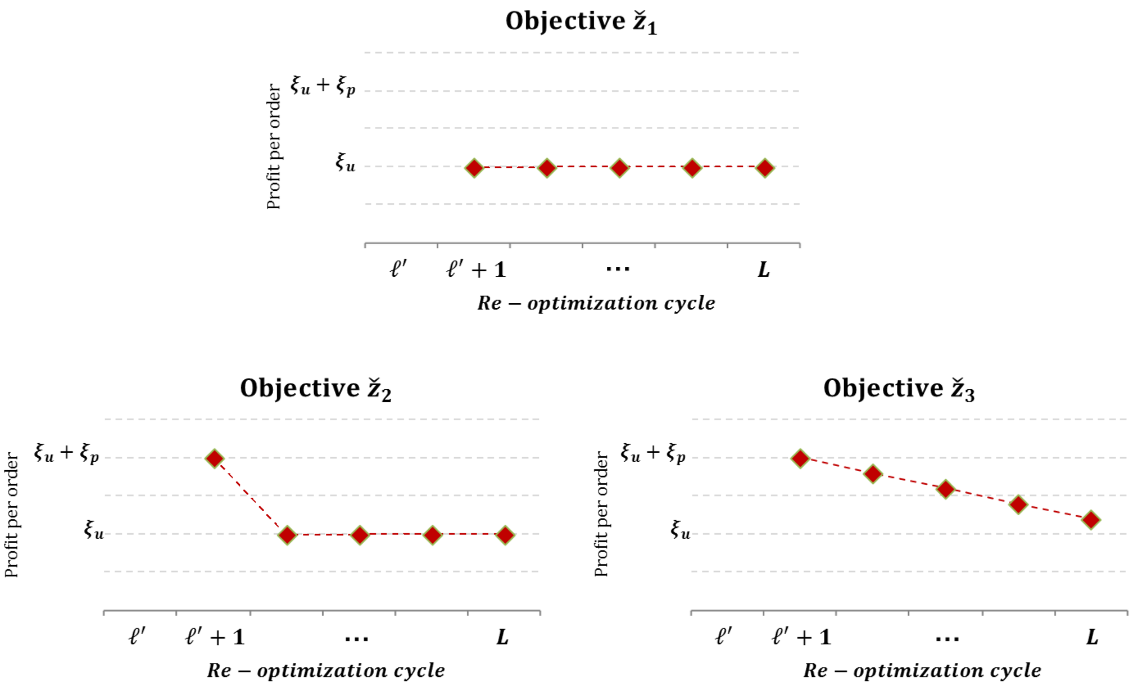
Προτείνονται τρεις (3) διαφορετικές αντικειμενικές συναρτήσεις, αναφορικά με το πρόβλημα που επιλύεται σε κάθε περίοδο αναδρομολόγησης:

α) Η αντικειμενική συνάρτηση $\check{z}_1$, η οποία αποτελείται από δύο όρους με λεξικογραφική δομή. Ο πρώτος όρος προσπαθεί να μεγιστοποιήσει την εξυπηρέτηση των ΔΑ, αναθέτοντας ένα σταθερό κέρδος σε κάθε ΔΑ που εξυπηρετείται (βλ. Π.15). Ο δεύτερο όρος ελαχιστοποιεί τα κόστη διαδρομής (για το μέγιστο αριθμό ΔΑ).

β) Η αντικειμενική συνάρτηση $\check{z}_2$ η οποία αποτελείται από τρεις λεξικογραφικούς όρους: ο πρώτος όρος μεγιστοποιεί την εξυπηρέτηση των ΔΑ (σταθερό κέρδος ανά ΔΑ), ο δεύτερος όρος προσδίδει επιπρόσθετο κέρδος σε κάθε απαίτηση (στατική ή δυναμική) η οποία εξυπηρετείται μέχρι τη χρονική στιγμή της επερχόμενης αναδρομολόγησης, και ο τρίτος όρος ελαχιστοποιεί τα κόστη διαδρομής.

γ) Τέλος, ορίζεται η αντικειμενική συνάρτηση $\check{z}_3$ η οποία τροποποιεί την $\check{z}_2$ αναφορικά με το επιπρόσθετο κέρδος (του δευτέρου όρου της $\check{z}_2$). Στη περίπτωση αυτή, το κέρδος αφορά τις απαιτήσεις που εξυπηρετούνται σε οποιαδήποτε μελλοντική περίοδο αναδρομολόγησης και μειώνεται γραμμικά ανάλογα με τη περίοδο εξυπηρέτησης της απαίτησης.

Έστω ξ_u το σταθερό κέρδος που ανατίθεται για κάθε εξυπηρετούμενη ΔΑ και ξ_p το επιπρόσθετο κέρδος σε περίπτωση που μία απαίτηση (στατική ή δυναμική) εξυπηρετείται μέχρι την επερχόμενη περίοδο αναδρομολόγησης. Με βάση την παραπάνω ορολογία, το κέρδος που ανατίθεται σε κάθε απαίτηση σε σχέση με τις τρεις προτεινόμενες αντικειμενικές συναρτήσεις αποτυπώνεται στο Σχήμα Π.4. Χρησιμοποιώντας τη κατάλληλη αντικειμενική ανά περίπτωση, η μέθοδος επίλυσης μπορεί να οδηγηθεί από την αποκλειστική μεγιστοποίηση των εξυπηρετούμενων ΔΑ (αντικειμενική συνάρτηση $\check{z}_1$), έως και τη μεγιστοποίηση της παραγωγικότητας των οχημάτων (αντικειμενικές συναρτήσεις $\check{z}_2$ και $\check{z}_3$).



Σχήμα Π.4. Το κέρδος ανά απαίτηση σε σχέση με τις τρεις προτεινόμενες αντικειμενικές συναρτήσεις

Για την υλοποίηση των αντικειμενικών συναρτήσεων $\check{z}_2$ και $\check{z}_3$, οι χρονικές στιγμές αναδρομολόγησης θα πρέπει να είναι προκαθορισμένες (και γνωστές εκ των προτέρων). Οι πολιτικές αναδρομολόγησης FTR που περιεγράφηκαν προηγουμένως είναι πιο κατάλληλες για τη περίπτωση αυτή. Πολιτικές που βασίζονται στο πλήθος των αφιχθέντων ΔΑ, μπορούν να υλοποιηθούν μόνο με την αντικειμενική $\check{z}_1$.

Τροποποιήσεις του ΠΔΔΟΠ για την εφαρμογή σε περιορισμένο στόλο οχημάτων

Για τη μοντελοποίηση του ΠΔΔΟΠΠΠ ως πρόβλημα διαμερισμού συνόλου (SPP), έγιναν οι παρακάτω τροποποιήσεις: α) ενσωμάτωση των νέων αντικειμενικών συναρτήσεων, β) εξασφάλιση πως κάθε στατική απαίτηση θα εξυπηρετηθεί (ακριβώς μία φορά), ενώ κάθε ΔΑ μπορεί να εξυπηρετηθεί το πολύ μία φορά, και γ) ενσωμάτωση του περιορισμού αναφορικά με τα διαθέσιμα οχήματα . Συνεπώς, η μοντελοποίηση του ΚΠ (set-partitioning problem), μετατρέπεται ως εξής:

Ελαχιστοποίηση $$\sum_{r\in\Omega'} \tilde{c}_r y_r \qquad (\Pi.20)$$

Υπό τους περιορισμούς: $$\sum_{r\in\Omega} e_{ir} y_r = 1 \qquad \forall i \in C \qquad (\Pi.21)$$

$$\sum_{r\in\Omega} e_{ir} y_r \le 1 \qquad \forall i \in F \qquad (\Pi.22)$$

$$\sum_{r \in \Omega_p} y_r \leq |K_d| \quad (\Pi.23)$$

$$y_r = \{0,1\} \qquad \forall r \in \Omega \quad (\Pi.24)$$

Η αντικειμενική συνάρτηση (Π.20) ελαχιστοποιεί το συνολικό κόστος των δρομολογίων (όπου $\tilde{c}_r$ είναι το τροποποιημένο κόστος, το οποίο συμπεριλαμβάνει τα κέρδη ξ_u και ξ_p). Ο περιορισμός (Π.21) εξασφαλίζει πως κάθε στατική απαίτηση θα εξυπηρετηθεί ακριβώς μία φορά από ένα όχημα, ενώ ο περιορισμός (Π.22) ορίζει πως κάθε ΔΑ μπορεί να εξυπηρετηθεί το πολύ μία φορά. Τέλος, ο περιορισμός (Π.23) περιορίζει τον αριθμό των διαθέσιμων οχημάτων.

Πειραματική διερεύνηση

Πειραματικές περιπτώσεις που χρησιμοποιήθηκαν

Στην παρούσα πειραματική διερεύνηση, χρησιμοποιήθηκαν τα σύνολα προβλημάτων R1 και C1 του Solomon (12 και 9 πειράματα, αντίστοιχα). Όπως και για το ΠΔΔΟΠ, χρησιμοποιήθηκαν επίσης τα πειράματα R100 και C100 (χωρίς χρονικά παράθυρα). Ο Πίνακας Π.2 συνοψίζει τις πειραματικές περιπτώσεις.

Επιπρόσθετα, εξετάστηκαν διαφορετικές τιμές αναφορικά με τη *διαθεσιμότητα των οχημάτων*, δηλαδή τον αριθμό των επιπλέον οχημάτων που είναι διαθέσιμα στο κέντρο διανομής για την εξυπηρέτηση ΔΑ. Για κάθε ένα από τα 23 παραπάνω πειράματα, εξετάστηκαν τρεις (3) διαφορετικές τιμές διαθέσιμων οχημάτων στο κέντρο διανομής, 0, 2 ή 4 οχήματα (εφεξής ορίζονται ως V-0, V-2 και V-4, αντίστοιχα). Συνεπώς, κατασκευάστηκαν 69 διαφορετικές περιπτώσεις (3 x 23). Για κάθε περίπτωση, θεωρήθηκε η περίπτωση μέτριου δυναμικού περιεχομένου (50% ΔΑ αναφορικά με το σύνολο των απαιτήσεων) και κατασκευάστηκαν 10 διαφορετικά προβλήματα (επιλέγοντας διαφορετικές στατικές απαιτήσεις). Συνεπώς, δημιουργήθηκαν 690 διαφορετικά προβλήματα.

Πίνακας Π.2. Πειράματα που χρησιμοποιήθηκαν για το ΠΔΔΟΠΠΠ

Γεωγραφική κατανομή	Χρονικά Παράθυρα	# Πειραμάτων	Πειράματα
Ομοιόμορφη	ΝΑΙ	12	R101, R102 ,R103, R104, R105, R106, R107, R108, R109, R110, R111, R112
Ομαδοποιημένη	ΝΑΙ	9	C101, C102, C103, C104, C105, C106, C107, C108, C109
Ομοιόμορφη	ΟΧΙ	1	R100
Ομαδοποιημένη	ΟΧΙ	1	C100

Αξιολόγηση της ευρετικής μεθόδου επίλυσης αναφορικά με τη βέλτιστη λύση

Αρχικά, εξετάζεται η απόδοση της ευρετικής μεθόδου επίλυσης σε περιβάλλον περιορισμένου στόλου οχημάτων, συγκριτικά με τη βέλτιστη λύση. Τα αποτελέσματα υποδεικνύουν πως η ευρετική μέθοδος εμφανίζει παρόμοια αποτελέσματα με τη περίπτωση του απεριόριστου στόλου. Συγκεκριμένα, στο περιβάλλον αυτό, η ευρετική μέθοδος αποκλίνει 1.9% από τη βέλτιστη λύση κατά μέσο όρο για όλα τα προβλήματα, παρέχοντας λύσεις σε ιδιαίτερα αποτελεσματικούς υπολογιστικούς χρόνους.

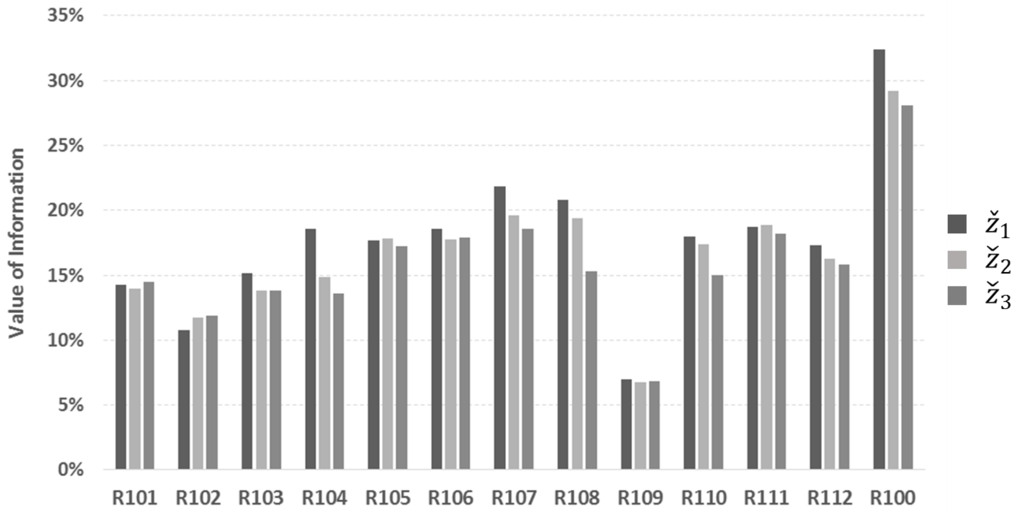
Αξιολόγηση της στρατηγικών αναδρομολόγησης

Στη συνέχεια εξετάζεται η συμπεριφορά των στρατηγικών αναδρομολόγησης σε περιβάλλον περιορισμένου στόλου οχημάτων, με κύριο στόχο την συσχέτιση της συμπεριφοράς με αυτή που παρατηρήθηκε στη περίπτωση του απεριόριστου στόλου. Προκειμένου να αποτελέσματα να είναι συγκρίσιμα, χρησιμοποιήθηκαν πολιτικές που βασίζονται στον αριθμό των ΔΑ (SRR, NRR-1, NRR-2 και NRR-3). Εφόσον οι περίοδοι αναδρομολόγησης δεν είναι προκαθορισμένες, η παρούσα ανάλυση χρησιμοποιεί την αντικειμενική συνάρτηση $\check{z}_1$. Η ανάλυση υποδεικνύει πως τα αποτελέσματα συμφωνούν με για τα αντίστοιχα της περίπτωσης απεριόριστου στόλου. Συγκεκριμένα, η στρατηγική SRR-PR παρέχει τα καλύτερα αποτελέσματα (ελάχιστο VoI) και η τακτική PR υπερισχύει της FR (κατά μέσο όρο) σε όλες τις περιπτώσεις. Παρόμοια συμπεριφορά παρατηρείται για τις διαφορετικές τιμές διαθεσιμότητας οχημάτων.

Αξιολόγηση των προτεινόμενων αντικειμενικών συναρτήσεων

Στην ενότητα αυτή, αξιολογείται η απόδοση των τριών προτεινόμενων αντικειμενικών συναρτήσεων που περιγράφηκαν ανωτέρω ($\check{z}_1$, $\check{z}_2$ και $\check{z}_3$). Η παρούσα διερεύνηση περιλαμβάνει όλα τα προβλήματα του συνόλου R1 (13 πειράματα, μαζί με το R100), για τις τρεις τιμές διαθεσιμότητας οχημάτων (V-0, V-2 και V-4) και χρησιμοποιώντας τα 10 διαφορετικά προβλήματα ανά πείραμα (390 προβλήματα στο σύνολο). Εφόσον οι αντικειμενικές συναρτήσεις $\check{z}_2$ και $\check{z}_3$ μπορούν να χρησιμοποιηθούν μόνο σε περιπτώσεις προκαθορισμένων περιόδων αναδρομολόγησης, χρησιμοποιήθηκαν οι πολιτικές FTR. Συγκεκριμένα, εξετάστηκαν τέσσερις (4) πολιτικές: FTR-10, FTR-20, FTR-40 και FTR-60 που αντιστοιχούν σε αναδρομολόγηση κάθε 10, 20, 40 και 60 μονάδες του T_{max} (230 μονάδες στα πειράματα), αντιστοίχως. Κάθε πολιτική εκτελέστηκε σε συνδυασμό με τις πολιτικές FR και PR (συνεπώς συνολικά 3,120 προβλήματα = 390x8).

Στο Σχήμα Π.6 παρουσιάζεται η απόδοση (αναφορικά με το VoI) της κάθε αντικειμενικής συνάρτησης για κάθε ένα από τα 13 πειράματα. Τα αποτελέσματα αφορούν τη μέση τιμή των διαφόρων στρατηγικών αναδρομολόγησης και προβλημάτων. Σύμφωνα με το Σχήμα, οι αντικειμενικές συναρτήσεις $\check{z}_2$ και $\check{z}_3$ (που λαμβάνουν υπόψη τη παραγωγικότητα των οχημάτων) καταλήγουν σε αποτελεσματικότερες λύσεις συγκριτικά με την $\check{z}_1$, κατά κύριο λόγο σε περιπτώσεις με μεγάλο εύρος χρονικών παραθύρων (R103, R104, R107, R108) ή σε περιπτώσεις χωρίς χρονικά παράθυρα (R100).

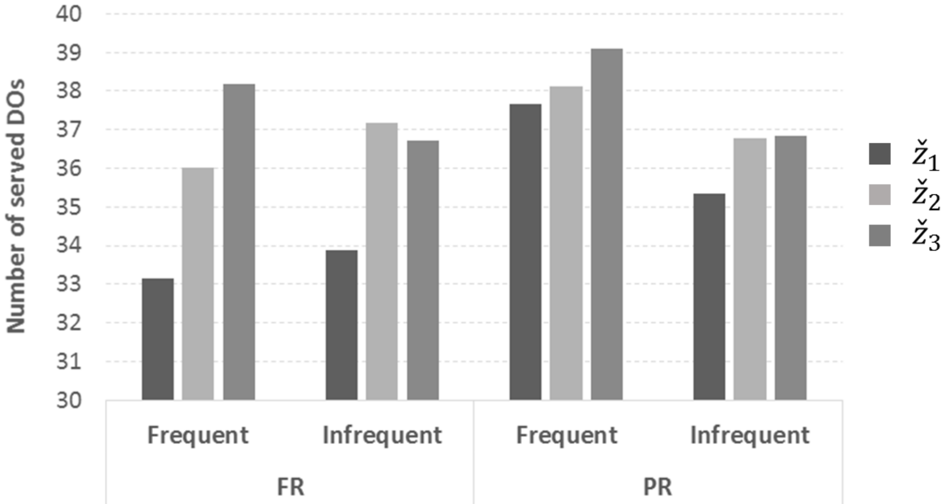


Σχήμα Π.6. Συνολική μέση απόδοση των τριών αντικειμενικών συναρτήσεων ανά πείραμα

Στην διατριβή διερευνήθηκε επίσης η συμπεριφορά των αντικειμενικών συναρτήσεων αναφορικά με τη συχνότητα αναδρομολόγησης. Η ανάλυση υποδεικνύει πως η αντικειμενική συνάρτηση $\check{z}_3$ επιτυγχάνει καλύτερα αποτελέσματα με χαμηλότερη συχνότητα αναδρομολόγησης (FTR-10, FTR-20) λόγω του ότι ευνοεί την ευέλικτη κατανομή των ΔΑ στην πλέον κατάλληλη περίοδο (χωρίς να εξαναγκάζει τις απαιτήσεις να εξυπηρετηθούν μέχρι την επερχόμενη περίοδο, όπως η $\check{z}_2$). Παρατηρείται επίσης πως η απόδοση των αντικειμενικών $\check{z}_2$ και $\check{z}_3$ συγκριτικά με την $\check{z}_1$ βελτιώνεται σημαντικά όταν η ακολουθείται η τακτική FR. Επιπρόσθετα, κατάλληλη ανάλυση αναφορικά με το εύρος των χρονικών παραθύρων και τη διαθεσιμότητα των οχημάτων, υποδεικνύει πως μέσω των αντικειμενικών συναρτήσεων $\check{z}_2$ και $\check{z}_3$ επιτυγχάνονται καλύτερα αποτελέσματα σε περιπτώσεις με διευρυμένα χρονικά παράθυρα (π.χ. >40% του T_{max}) και σχετικά υψηλή διαθεσιμότητα οχημάτων. Αντίθετα, σε περιπτώσεις στενών (περιορισμένων) χρονικών παραθύρων και περιορισμένης διαθεσιμότητας οχημάτων, αντικειμενικές συναρτήσεις που λαμβάνουν υπόψη τους τη παραγωγικότητα των οχημάτων, δεν επιφέρουν σημαντική βελτίωση στις αντίστοιχες λύσεις.

Για την αρτιότερη αποτύπωση της απόδοσης των αντικειμενικών συναρτήσεων, στο Σχήμα Π.7 παρουσιάζεται η απόδοση των συναρτήσεων αυτών αναφορικά με τον αριθμό των εξυπηρετημένων ΔΑ κάτω από παραμέτρους που ευνοούν τις $\check{z}_2$ και $\check{z}_3$. Συγκεκριμένα, α) τιμή

V-4 για τη διαθεσιμότητα οχημάτων, και β) πειράματα με μεγάλα ή καθόλου χρονικά παράθυρα (R104, R108 και R100). Από το Σχήμα αυτό διαφαίνεται πως οι εν λόγω αντικειμενικές συναρτήσεις καταλήγουν σε λύσεις στις οποίες εξυπηρετούνται έως και 15% περισσότερες ΔΑ (για συχνή αναδρομολόγηση με τακτική FR).

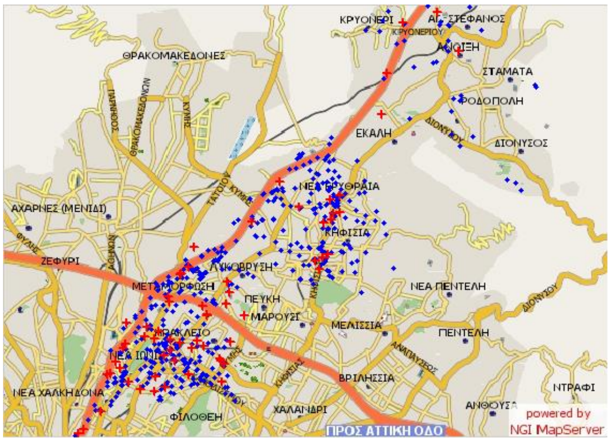


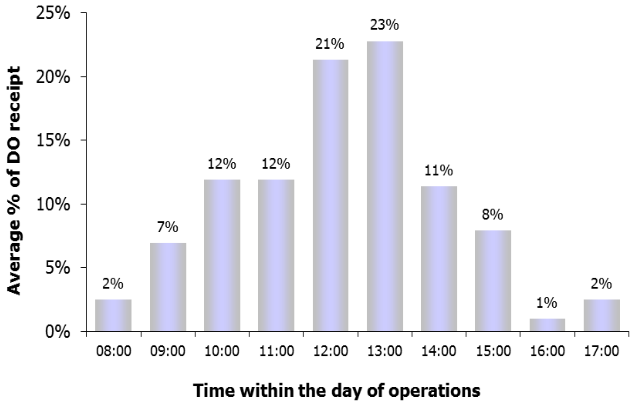
Σχήμα Π.7. Συνολική μέση απόδοση των τριών αντικειμενικών ανά πείραμα

Μελέτη περίπτωσης σε πραγματικό περιβάλλον εταιρείας ταχυμεταφορών

Η προτεινόμενη μέθοδος επίλυσης του ΠΔΔΟΠΠΠ εφαρμόστηκε σε περιβάλλον της εταιρείας Ταχυμεταφορές ΕΛΤΑ Α.Ε., η οποία κατέχει το τρίτο μεγαλύτερο μερίδιο αγοράς στην Ελλάδα. Για τη μελέτη αυτή, χρησιμοποιήθηκαν πραγματικά δεδομένα της εταιρείας και εφαρμόστηκε η προτεινόμενη B&P μέθοδος για τη δρομολόγηση των ΔΑ. Τα αποτελέσματα συγκρίνονται με: α) τις χειρωνακτικές πρακτικές των διακινητών της εταιρείας, καθώς και β) με τα αποτελέσματα προηγμένου αλγορίθμου παρεμβολής.

Η μελέτη περίπτωσης έλαβε χώρα σε κέντρο διανομής (ΚΔ) που καλύπτει συγκεκριμένη περιοχή της Αθήνας (700 km^2). Το ΚΔ εξυπηρετεί κατά μέσο όρο 450 στατικές απαιτήσεις (ΣΑ) και 70 δυναμικές απαιτήσεις (ΔΑ) ανά ημέρα με στόλο 13 οχημάτων. Τα δεδομένα συλλέχθηκαν για περίοδο τριών ημερών (477, 491 και 370 στατικές απαιτήσεις, καθώς και 68, 68 και 66 ΔΑ ανά ημέρα, αντίστοιχα). Το Σχήμα Π.8a αποτυπώνει τις θέσεις των απαιτήσεων (πελατών) για μία ημέρα και το Σχ. Π.8b τη χρονική κατανομή των ΔΑ σε συνάρτηση με την ώρα της ημέρας.





(a) (b)

Σχήμα Π.8. (a) Αποτύπωση των απαιτήσεων μίας ημέρας σε ψηφιακό χάρτη (οι μπλε κύκλοι αντιπροσωπεύουν τις ΣΑ και οι κόκκινοι σταυροί τις ΔΑ); (b) κατανομή των ΔΑ σε σχέση με την ώρα τις ημέρας (όλες οι τρεις ημέρες)

Για τη πειραματική διερεύνηση, χρησιμοποιήθηκαν τέσσερα βασικά εργαλεία δρομολόγησης, όπως φαίνεται στον Πίνακα Π.3. Ο συνδυασμός διαφορετικών εργαλείων για τη δρομολόγηση των στατικών απαιτήσεων και των ΔΑ παρέχει πέντε διαφορετικά σενάρια (βλ. Πίνακα Π.4).

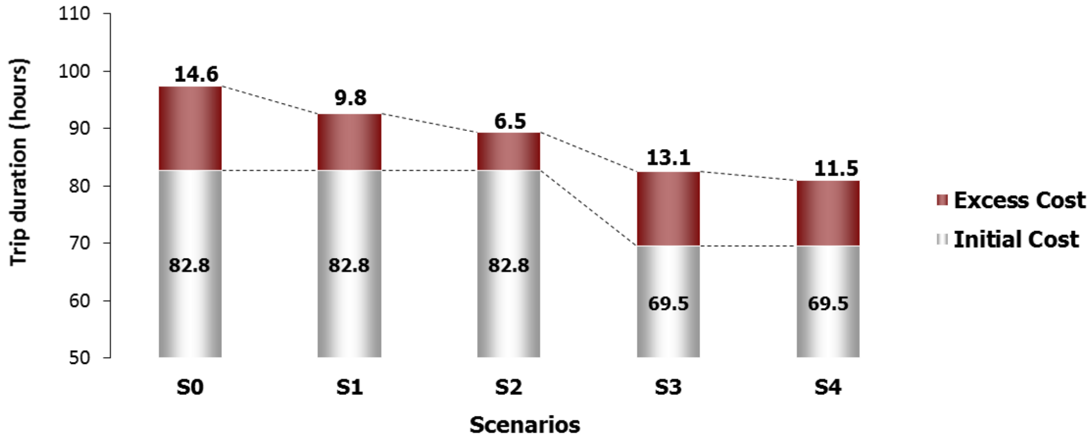
Πίνακας Π.3. Εργαλεία δρομολόγησης που εμπλέκονται στη πειραματική διερεύνηση

Εργαλεία	Δρομολόγηση για	Περιγραφή
Manual	ΣΑ & ΔΑ	Οι παρούσες χειρωνακτικές διαδικασίες που ακολουθούνται από τους διακινητές. Συμπεριλαμβάνει δρομολόγηση για στατικές απαιτήσεις και ΔΑ
SW	ΣΑ	Αρχικός προγραμματισμός των στατικών απαιτήσεων μέσω εμπορικής εφαρμογής δρομολόγησης
HEUR	ΔΑ	Ο προηγμένος αλγόριθμος παρεμβολής (insertion heuristic)
B&P	ΔΑ	Η προτεινόμενη B&P μέθοδος

Πίνακας Π.4. Σενάρια δρομολόγησης

Σενάριο	**Δρομολόγηση ΣΑ**		**Δρομολόγηση ΔΑ**		
	Manual	**SW**	**Manual**	**HEUR**	**B&P**
S0	✓		✓		
S1	✓			✓	
S2	✓				✓
S3		✓		✓	
S4		✓			✓

Στο Σχήμα Π.9 παρουσιάζεται η απόδοση (μέσος όρος για τις τρεις ημέρες) των πέντε σεναρίων αναφορικά με τη χρονική διάρκεια των δρομολογίων (σε ώρες). Από το Σχήμα διακρίνεται πως το σενάριο S4 παρουσιάζει τα καλύτερα αποτελέσματα από όλα τα υπόλοιπα σενάρια, με μέση βελτίωση της τάξεως του 16% αναφορικά με το σενάριο S0. Αναφορικά με τη διαχείριση των ΔΑ, η B&P μέθοδος (S2) υπερισχύει του αλγορίθμου παρεμβολής (S1) κατά 33.8%. Η βελτίωση αυτή μειώνεται στο 12.2% όταν η εμπορική εφαρμογή χρησιμοποιείται για την δρομολόγηση των ΣΑ.



Σχήμα Π.9. Συνολική απόδοση των σεναρίων δρομολόγησης (μέσος όρος όλων των ημερών)

Η ΠΕΡΙΠΤΩΣΗ ΜΕΤΑΦΟΡΤΩΣΗΣ ΣΤΟ ΠΔΔΟΠ

Οι προηγούμενες αναλύσεις του ΠΔΔΟΠ (απεριόριστου και περιορισμένου αριθμού οχημάτων) θεωρούσαν ως σταθερή την αρχική ανάθεση των στατικών απαιτήσεων (παραδόσεων) στα οχήματα. Ωστόσο, η διατήρηση αυτής της αρχικής ανάθεσης ενδέχεται να περιορίσει τη απόδοση του συστήματος, εφόσον οι αλλαγές του πλάνου που προκαλούνται από την άφιξη νέων ΔΑ μπορεί να προσδώσουν πλεονέκτημα σε τυχόν αλλαγές των στατικών απαιτήσεων μεταξύ των οχημάτων. Με βάση αυτή την παρατήρηση, στη παρούσα ενότητα εξετάζεται και επιλύεται μία καινοτόμα παραλλαγή του ΠΔΔΟΠ η οποία επιτρέπει μεταφορτώσεις κατά τη διάρκεια εκτέλεσης των δρομολογίων. Το πρόβλημα αυτό αναφέρεται ως το *ΠΔΔΟΠ με Μεταφορτώσεις* (ΠΔΔΟΠΜΦ). Επιτρέποντας τέτοιου είδους αλλαγές, αναμένεται η αποτελεσματικότερη εκμετάλλευση του στόλου, ανά-κατανέμοντας το φόρτο εργασίας όπως απαιτείται βάσει της δυναμικότητας του συστήματος. Οι μεταφορτώσεις μπορούν να πραγματοποιηθούν επιτρέποντας στα οχήματα να συναντηθούν σε πραγματικό χρόνο. Η τακτική αυτή είναι ιδιαίτερα συνηθισμένη σε εταιρείες ταχυμεταφορών και χρηματαποστολών.

Για την αντιμετώπιση του ΠΔΔΟΠΜΦ με απεριόριστο πλήθος οχημάτων, μοντελοποιείται αρχικά το σχετικό πρόβλημα αναδρομολόγησης και συγκρίνεται η βέλτιστη λύση του με το ΠΔΔΟΠ (που δεν επιτρέπει μεταβιβάσεις). Στη συνέχεια, αναπτύσσεται πλαίσιο ευρετική επίλυσης για την αντιμετώπιση προβλημάτων πρακτικού μεγέθους. Το πλαίσιο αυτό χρησιμοποιείται για την επίλυση του συνολικού προβλήματος (πολλαπλοί περίοδοι) και εξετάζεται η επίδραση διαφορετικών στρατηγικών αναδρομολόγησης στη ποιότητα της λύσης.

Το πρόβλημα αναδρομολόγησης στο ΠΔΔΟΠΜΦ

Επιτρέποντας μεταφορτώσεις κατά την επίλυση του προβλήματος αναδρομολόγησης, ενδέχεται να αυξηθεί σημαντικά η πολυπλοκότητα του συστήματος. Από διοικητικής απόψεως, ενδέχεται να μην είναι πρακτικό να επιτρέπονται πολλαπλές μεταφορτώσεις ανά απαίτηση, ή μεταφορτώσεις μεταξύ άνω των δύο οχημάτων, εφόσον οι πρακτικές αυτές ενδέχεται να αποφέρουν σύγχυση στους οδηγούς και επιπρόσθετη διαχείριση. Υπολογίζοντας τα παραπάνω, το πρόβλημα αναδρομολόγησης για το ΠΔΔΟΠΜΦ επιλύεται λαμβάνοντας υπόψη τις παρακάτω παραδοχές:

a) Όλες οι απαιτήσεις πρέπει να εξυπηρετηθούν (στατικές επιδόσεις και ΔΑ)
b) Κάθε όχημα μπορεί να συμμετέχει σε μία μόνο μεταφόρτωση ανά επίλυση του προβλήματος αναδρομολόγησης.
c) Αναφορικά με τα σημεία μεταφόρτωσης, εξετάζονται δύο περιπτώσεις όπου η μεταφόρτωση λαμβάνει χώρα: i) σε προκαθορισμένα σημεία (γνωστά εκ των προτέρων), ή ii) στις τοποθεσίες όλων των μη εξυπηρετούμενων στατικών και δυναμικών απαιτήσεων (συμπεριλαμβανομένων και των θέσεων των οχημάτων).

Μαθηματική μοντελοποίηση του προβλήματος αναδρομολόγησης στο ΠΔΔΟΠΜΦ

Η παρακάτω μοντελοποίηση βασίστηκε στην εργασία του Cortes *et al.* (2010), η οποία αναλύει μεταφορτώσεις για τη περίπτωση του Προβλήματος Παραλαβών και Επιδόσεων (στο οποίο κάθε απαίτηση σχετίζεται με μία τοποθεσία παραλαβής και μία επίδοσης).

<u>Βασικές παραδοχές μοντελοποίησης</u>

Στην ανάπτυξη του μοντέλου χρησιμοποιήθηκαν οι παρακάτω παραδοχές αναφορικά με τα σημεία μεταφόρτωσης: α) κάθε τέτοιο σημείο u αποτελείται από δύο κόμβους $s(u)$ και $f(u)$, οι οποίοι σχετίζονται με την έναρξη και ολοκλήρωση της μεταφόρτωσης, β) δημιουργούνται κλώνοι των συνόλων των μη εξυπηρετούμενων απαιτήσεων (N') και των θέσεων των οχημάτων (M'), γ) κάθε τοποθεσία μη εξυπηρετούμενου πελάτη χαρακτηρίζεται πλέον από τρεις κόμβους: τον αρχικό κόμβο $i \in (N \cup M \cup 0)$, τον κόμβο έναρξης μεταφόρτωσης $s(u)$, και τον κόμβο ολοκλήρωσης της μεταφόρτωσης $f(u)$, όπου $u \in (N' \cup M' \cup 0')$ δηλώνει το σημείο μεταφόρτωσης που αντιστοιχεί στον κόμβο $i \in (N \cup M \cup 0)$.

<u>Το μαθηματικό μοντέλο</u>

Επιπρόσθετα από τον συμβολισμό που παρατέθηκε στο ΠΔΔΟΠ προηγουμένως, ορίζεται το σύνολο όλων των σημείων μεταφόρτωσης $U = U_f \cup \{0'\} \cup M' \cup N'$, όπου U_f αντιπροσωπεύει το σύνολο των προκαθορισμένων σημείων. Βάσει αυτού, ορίζουμε το σύνολο των κόμβων του μοντέλου ως $W = N \cup M \cup \{0\} \cup s(U) \cup f(U)$. Το σύνολο των ακμών A ορίζεται επίσης κατάλληλα, έτσι ώστε ενώσεις κόμβων μη σχετικές με το πρόβλημα δε συμπεριλαμβάνονται.

Το μοντέλο περιλαμβάνει τρία διαφορετικά σύνολα μεταβλητών: i) τη μεταβλητή x_{ijk} που ισούται με 1 αν το όχημα $k \in V$ διανύει την ακμή $(i,j) \in A$ και μηδέν σε άλλη περίπτωση, ii) τη μεταβλητή w_{ik}, που αντιπροσωπεύει το χρόνο έναρξης εξυπηρέτησης της απαίτησης (κόμβου) $i \in W$ από το όχημα $k \in V$, όπου $w_{s(u)k}$ και $w_{f(u)k}$ αντιστοιχούν στον χρόνο άφιξης και αναχώρησης από το σημείο μεταφόρτωσης, αντίστοιχα, και iii) τη μεταβλητή z_j^{ki} η οποία χρησιμοποιείται για την ιχνηλάτηση της κατάστασης του φορτίου κάθε απαίτησης όταν ταξιδεύει από κόμβο σε κόμβο. Η μεταβλητή λαμβάνει τη τιμή 1 αν η απαίτηση $i \in N$ υπάρχει στο όχημα $k \in K$ όταν φτάνει στον κόμβο $j \in W \backslash M$ και 0 διαφορετικά, για κάθε $i \in N, k \in K$.

Αντικειμενικός στόχος του προβλήματος είναι η ελαχιστοποίηση του συνολικού κόστους δρομολόγησης καθ' όλο το εύρος του χρονικού ορίζοντα $[\mathrm{T}_\ell, \mathrm{T}_{\max}]$ και δίνεται από τη συνάρτηση (Π.25) παρακάτω:

$$min(z) = \sum_{k \in V} \sum_{(i,j) \in A} c_{ijk} x_{ijk} \qquad \text{(Π.25)}$$

Περιορισμοί δρομολογίων

$$\sum_{j \in W \backslash (M \cup f(U))} x_{\mu_k jk} = 1 \qquad \forall k \in K \qquad \text{(Π.26)}$$

$$\sum_{i \in N \cup \{\mu_k\} \cup f(U)} x_{i0k} = 1 \qquad \forall k \in K \qquad \text{(Π.27)}$$

$$\sum_{i \in W \backslash (\{0\} \cup s(U))} x_{ihk} - \sum_{j \in W \backslash (M \cup f(U))} x_{hjk} = 0 \qquad \forall k \in K, \forall h \in N \qquad \text{(Π.28)}$$

$$\sum_{i \in N \cup \{\mu_k\}} x_{is(u)k} - x_{s(u)f(u)k} = 0 \qquad \forall k \in K, \forall u \in U \qquad \text{(Π.29)}$$

$$\sum_{j \in N \cup \{0\}} x_{f(u)jk} - x_{s(u)f(u)k} = 0 \qquad \forall k \in K, \forall u \in U \qquad \text{(Π.30)}$$

Περιορισμοί απαιτήσεων

$$\sum_{k \in K} \sum_{j \in W \backslash (M \cup f(U))} x_{ijk} = 1 \qquad \forall i \in N \qquad \text{(Π.31)}$$

Χρονικοί περιορισμοί

$$x_{\mu_k ik} = 1 \implies w_{ik} \geq t_{\mu_k i} \qquad \forall k \in K, \forall i \in N \cup 0 \qquad \text{(Π.32)}$$

$$x_{\mu_k s(u)k} = 1 \implies w_{s(u)k} \geq t_{\mu_k s(u)} \qquad \forall k \in K, \forall u \in U \qquad \text{(Π.33)}$$

$$x_{ijk} = 1 \implies w_{jk} \geq w_{ik} + t_{ij} + s_i \qquad \forall k \in K, \forall (i,j) \in \{(i,j): i \in N, j \in N \cup 0\} \qquad \text{(Π.34)}$$

$$x_{is(u)k} = 1 \implies w_{s(u)k} \geq w_{ik} + t_{is(u)} + s_i \quad \forall k \in K, \forall i \in N, \forall u \in U \quad (\Pi.35)$$

$$x_{s(u)f(u)k} = 1 \implies w_{f(u)k} \geq w_{s(u)k} + t_{s(u)f(u)} \quad \forall k \in K, \forall u \in U \quad (\Pi.36)$$

$$x_{f(u)jk} = 1 \implies w_{jk} \geq w_{f(u)k} + t_{f(u)j} \quad \forall k \in K, \forall j \in N \cup 0, \forall u \in U \quad (\Pi.37)$$

$$x_{f(u)s(\varphi)k} = 1 \implies w_{s(\varphi)k} \geq w_{f(u)k} + t_{f(u)s(\varphi)} \quad \forall k \in K, \forall u \in U, \forall \varphi \in U \backslash \{u\} \quad (\Pi.38)$$

<u>Περιορισμοί ροής απαιτήσεων</u>

$$\sum_{k \in K} \sum_{i \in F} z_{\mu_k}^{ki} = \sum_{k \in K} \sum_{i \in C_k} z_{\mu_k}^{ki} - |C| = 0 \quad (\Pi.39)$$

$$\sum_{k \in K} \sum_{i \in C} z_0^{ki} = \sum_{k \in K} \sum_{i \in F} z_0^{ki} - |F| = 0 \quad (\Pi.40)$$

$$x_{hjk} = 1 \implies z_h^{ki} = z_j^{ki} \quad \forall k \in K, \forall i \in N, \forall (h,j) \in A^{U}\text{[1]} \textit{ such that } h \neq i \quad (\Pi.41)$$

$$x_{ijk} = 1 \implies z_i^{ki} - z_j^{ki} = 1 \quad \forall k \in K, \forall i \in C, \forall j \in W \backslash (M \cup f(U)) \quad (\Pi.42)$$

$$x_{ijk} = 1 \implies z_j^{ki} - z_i^{ki} = 1 \quad \forall k \in K, \forall i \in F, \forall j \in W \backslash (M \cup f(U)) \quad (\Pi.43)$$

$$\sum_{k \in K} z_{s(u)}^{ki} - \sum_{k \in K} z_{f(u)}^{ki} = 0 \quad \forall u \in U, \forall i \in N \quad (\Pi.44)$$

$$z_{s(u)}^{ki} + z_{f(u)}^{mi} = 2 \implies w_{f(u)m} \geq w_{s(u)k} + \tilde{\Xi} \quad \forall u \in U, \forall k, m \in K, k \neq m, \forall i \in N \quad (\Pi.45)$$

$$z_{s(u)}^{ki} + z_{f(u)}^{mi} = 2 \implies w_{f(u)k} \geq w_{s(u)m} \quad \forall u \in U \backslash U_f \cup \{0'\}, \forall i \in N, \forall k, m \in K, k \neq m \quad (\Pi.46)$$

<u>Επιχειρησιακοί περιορισμοί</u>

$$\sum_{r \in U} \sum_{k \in K} z_{s(u)}^{ki} \leq 1 \quad \forall i \in N \quad (\Pi.47)$$

$$\sum_{i \in W \backslash f(U)} \sum_{u \in U} x_{is(u)k} \leq 1 \quad \forall k \in K \quad (\Pi.48)$$

$$max(a_i, T) \sum_{j \in W \backslash (M \cup f(U))} x_{ijk} \leq w_{ik} \leq b_i \sum_{j \in W \backslash (M \cup f(U))} x_{ijk} \quad \forall k \in K, \forall i \in N \quad (\Pi.49)$$

$$\sum_{i \in N} q_i z_j^{ki} \leq \bar{Q} \quad \forall j \in N, \forall k \in K \quad (\Pi.50)$$

Οι περιορισμοί (Π.26) και (Π.27) εξασφαλίζουν πως τα οχήματα θα εκκινήσουν από τις παρούσες θέσεις τους και θα καταλήξουν στο κέντρο διανομής. Ο περιορισμός (Π.28) διασφαλίζει τη διατήρηση ροής των κόμβων του συνόλου N, ενώ οι περιορισμοί (Π.29) και (Π.30) διατηρούν τη ροή στους κόμβους μεταφόρτωσης. Επισημαίνεται πως οι περιορισμοί αυτοί επιτρέπουν στα οχήματα να φτάσουν στους κόμβους μεταφόρτωσης το πολύ μία φορά.

Οι περιορισμοί (Π.31) ορίζουν πως όλες οι απαιτήσεις θα εξυπηρετηθούν ακριβώς μία φορά. Οι περιορισμοί (Π.32) – (Π.38) διασφαλίζουν τη χρονική εφικτότητα ενός δρομολογίου και χρησιμοποιούνται για να εξαλείψουν κυκλικές διαδρομές (subtours). Επισημαίνεται επίσης πως για τον περιορισμό (Π.36), ο χρόνος διαδρομής μεταξύ του αρχικού και τελικού κόμβου του

[1] $A^U = A \setminus \{(s(u), f(u) | u \in U)\}$

σημείου μεταφόρτωσης $t_{s(u)f(u)}$, θεωρείται ως πολύ μικρός θετικός αριθμός προς αποφυγή κυκλικών διαδρομών με μηδενικό κόστος.

Οι περιορισμοί (Π.39) και (Π.40) ορίζουν τις αρχικές και τελικές συνθήκες φόρτωσης, αντίστοιχα. Συγκεκριμένα, ένα όχημα $k \in K$ ξεκινά από τη παρούσα θέση του μεταφέροντας ήδη τις απαιτήσεις C που του έχουν ανατεθεί και καταλήγει στο κέντρο διανομής μόνο με απαιτήσεις F. Ο περιορισμός (Π.41) εξασφαλίζει τη συνέχεια των φορτίων. Ένα φορτίο εκφορτώνεται μόνο στην αντίστοιχη θέση της απαίτησης (το φορτίο της απαίτησης $i \in N$ θα βρίσκεται στο όχημα όταν φτάσει στην θέση της απαίτησης $j \in N$ αν αυτό ήταν επίσης στο όχημα όταν το όχημα ήταν στην θέση της προηγούμενης απαίτησης $h \in N$). Ο περιορισμός (Π.42) εξασφαλίζει πως μία απαίτηση επίδοσης εκφορτώνεται μόνο όταν φτάσει στην αντίστοιχη θέση της. Ομοίως, ο περιορισμός (Π.43) διασφαλίζει πως μία απαίτησης παραλαβής μπορεί να φορτωθεί μόνο στην αντίστοιχη θέση της. Ο περιορισμός (Π.44) ορίζει τη διατήρηση ροής των μεταβλητών φορτίου και εξασφαλίζει πως όταν μία απαίτηση φτάσει σε ένα σημείο μεταφόρτωσης με οποιοδήποτε όχημα, τότε θα πρέπει να εγκαταλείψει το σημείο αυτό με οποιοδήποτε όχημα (ουσιαστικά, με το ίδιο ή με το άλλο όχημα του ζεύγους). Ο περιορισμός (Π.45) ορίζει πως αν μία απαίτηση φτάσει σε ένα σημείο μεταφόρτωσης με το όχημα $k_1 \in K$ και αναχωρήσει από το σημείο αυτό με το όχημα $k_2 \in K, k_2 \neq k_1$, τότε το όχημα k_1 θα πρέπει να φτάσει στο σημείο μεταφόρτωσης πριν από την αναχώρηση του οχήματος k_2 από το σημείο αυτό. $\tilde{\Xi}$ είναι μία τιμή η οποία αντιπροσωπεύει το χρόνο που απαιτείται για την απαίτηση να παραμείνει στο σημείο μεταφόρτωσης. Επιπρόσθετα, ο περιορισμός (Π.46) είναι παρόμοιος με τον (Π.45), αλλά εξασφαλίζει την ταυτόχρονη παρουσία των οχημάτων στο σημείο μεταφόρτωσης για τις περιπτώσεις που η μεταφόρτωση πραγματοποιείται σε κόμβους πελατών.

Αναφορικά με τους επιχειρησιακούς περιορισμούς, ο περιορισμός (Π.47) περιορίζει τον αριθμό των μεταβιβάσεων απαίτησης (το πολύ μία φορά), ενώ ο περιορισμός (Π.48) περιορίζει τον αριθμό των μεταβιβάσεων ανά όχημα. Τέλος, ο περιορισμός (Π.49) εξασφαλίζει πως κάθε απαίτηση εξυπηρετείται εντός του αντίστοιχου χρονικού παραθύρου και ο περιορισμός (Π.50) ορίζει πως το φορτίο κάθε οχήματος δε θα ξεπεράσει τη χωρητικότητά ($\bar{Q}$) του οχήματος.

Πλαίσιο επίλυσης του προβλήματος αναδρομολόγησης με ανταλλαγές φορτίων

Το προηγούμενο μοντέλο μπορεί να επιλυθεί με τη χρήση εμπορικής εφαρμογής (π.χ. CPLEX) για περιπτώσεις περιορισμένου μεγέθους. Για την επίλυση προβλημάτων πρακτικού μεγέθους, προτείνεται πλαίσιο ευρετικής επίλυσης (*Load Transfer Algorithm, LTA*). Συγκεκριμένα, χρησιμοποιείται κατάλληλη διαδικασία η οποία αναγνωρίζει ζεύγη οχημάτων τα οποία ενδέχεται να επωφεληθούν από τη μεταφόρτωση (η αναγνώριση ζευγών βασίζεται στη σχετική παραδοχή). Στη συνέχεια, για κάθε υποψήφιο ζεύγος που έχει αναγνωριστεί, χρησιμοποιείται κατάλληλος αλγόριθμος, ο οποίος επιλύει το πρόβλημα μεταφόρτωσης φορτίων.

Πειραματική διερεύνηση του ΠΔΔΟΠ με ανταλλαγές φορτίων

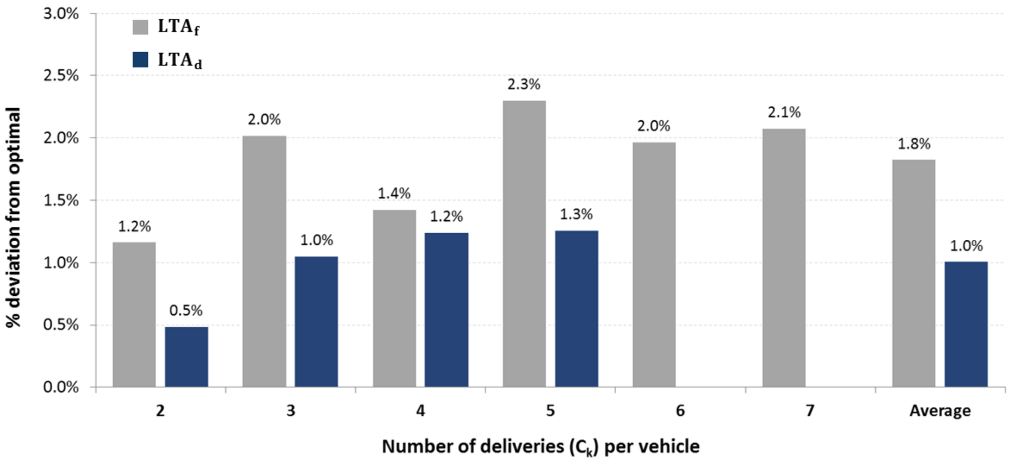
Αξιολόγηση της ευρετικής μεθόδου για ένα ζεύγος οχημάτων (LTA^p)

Για την αξιολόγηση της ευρετικής μεθόδου, κατασκευάστηκαν τυχαία προβλήματα με διαφορετικές τιμές βασικών παραμέτρων (όπως φαίνεται στον Πίνακα Π.5). Το σύνολο των προβλημάτων αυτών περιλαμβάνει 360 προβλήματα.

Πίνακας Π.5. Παράμετροι πειραματικής διερεύνησης

Παράμετρος	Περιγραφή	Τιμές (επίπεδα παραμέτρου)	# επιπέδων
C_k	Επιδόσεις ανά όχημα	2, …, 7	6
F	Δυναμικές απαιτήσεις	2, …, 7	6
$\wp$	Διαφορετικά προβλήματα	1,…,10	10

Κάθε ένα από τα προβλήματα επιλύθηκε αρχικά με τη συμβατική μέθοδο B&P που περιεγράφηκε για το ΠΔΔΟΠ, έτσι ώστε να ενσωματωθούν οι ΔΑ στο πλάνο, χωρίς μεταφορτώσεις (εφεξής θα αναφέρεται ως NTA). Στη συνέχεια τα προβλήματα επιλύθηκαν επιτρέποντας μεταφορτώσεις: α) βέλτιστα, με την επίλυση του μαθηματικού μοντέλου (OPT), καθώς και β) με τη χρήση του ευρετικού αλγορίθμου. Σε κάθε περίπτωση, εξετάστηκαν οι περιπτώσεις μεταφόρτωσης σε προκαθορισμένο σημείο ($\mathrm{LTA}_\mathrm{f}^\mathrm{opt}$ για τη βέλτιστη και LTA_f για την ευρετική μέθοδο), καθώς και μεταφόρτωση στις θέσεις όλων των μη εξυπηρετούμενων απαιτήσεων (αντίστοιχα, $\mathrm{LTA}_d^\mathrm{opt}$ και LTA_d).



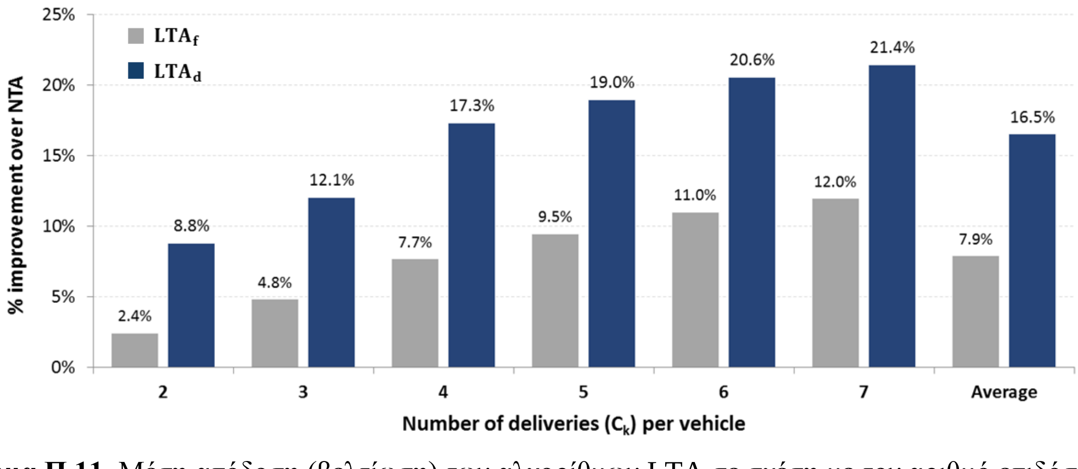
Σχήμα Π.10. Συνολική αξιολόγηση της ευρετικής μεθόδου συγκριτικά με τη βέλτιστη

Στο Σχήμα Π.10 παρουσιάζεται η απόκλιση της ευρετικής λύσης από τη βέλτιστη. Συμπεραίνεται ότι η ευρετική μέθοδος παρέχει ιδιαίτερα αποτελεσματικές λύσεις που έχουν απόκλιση κατά μέσο όρο της τάξης του 1.8% για το προκαθορισμένο σημείο μεταφόρτωσης και 1.0% για όλες τις θέσεις των πελατών.

Αναδρομολόγηση με μεταφόρτωση για δύο οχήματα

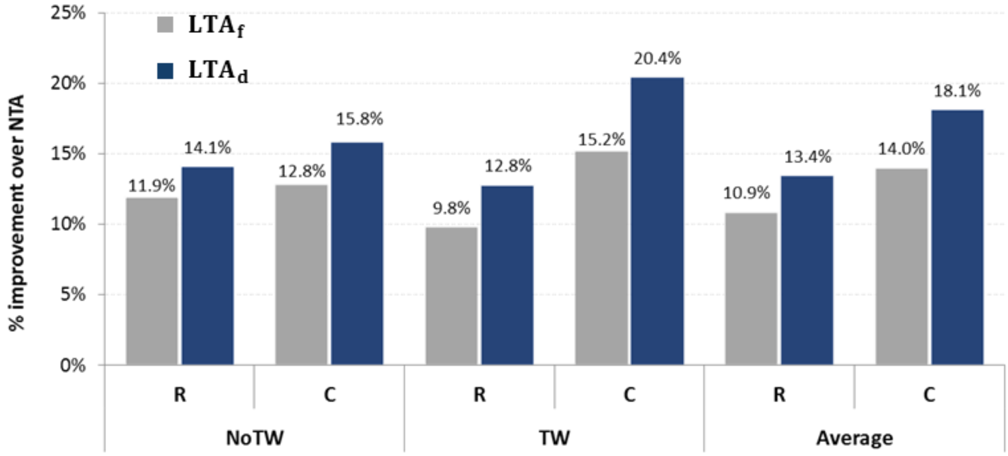
Στη παρούσα διερεύνηση που επικεντρώνεται σε ένα μόνο ζεύγος οχημάτων εξετάζουμε δύο σχετικά επιχειρησιακά σενάρια: α) τη περίπτωση όπου και τα δύο οχήματα είναι καθοδόν, και β) τη περίπτωση όπου ένα από τα δύο οχήματα βρίσκεται στο κέντρο διανομής. Για τη πρώτη περίπτωση, χρησιμοποιούμε τα αποτελέσματα από την ανάλυση της ευρετικής μεθόδου που παρουσιάστηκε προηγουμένως. Για τη δεύτερη περίπτωση, χρησιμοποιούμε κατάλληλα πειράματα από τα σύνολα προβλημάτων R1,C1, R2 και C2 του Solomon (1987), τα οποία τροποποιούνται ανάλογα έτσι ώστε να αντικατοπτρίζουν τα σενάριο προς διερεύνηση.

Στο Σχήμα Π.11 παρουσιάζεται η απόδοση των LTA_f και LTA_d ως ποσοστιαία διαφορά από την NTA για τη περίπτωση όπου τα δύο οχήματα είναι καθοδόν. Είναι διακριτό πως οι αλγόριθμοι LTA_f και LTA_d υπερέχουν σημαντικά της NTA, με βελτιώσεις της τάξεως του 7.9% και 16.5%, αντίστοιχα, κατά μέσο όρο. Η απόδοση φαίνεται να αυξάνεται με την αύξηση του αριθμού των επιδόσεων ανά όχημα.



Σχήμα Π.11. Μέση απόδοση (βελτίωση) των αλγορίθμων LTA σε σχέση με τον αριθμό επιδόσεων ανά όχημα

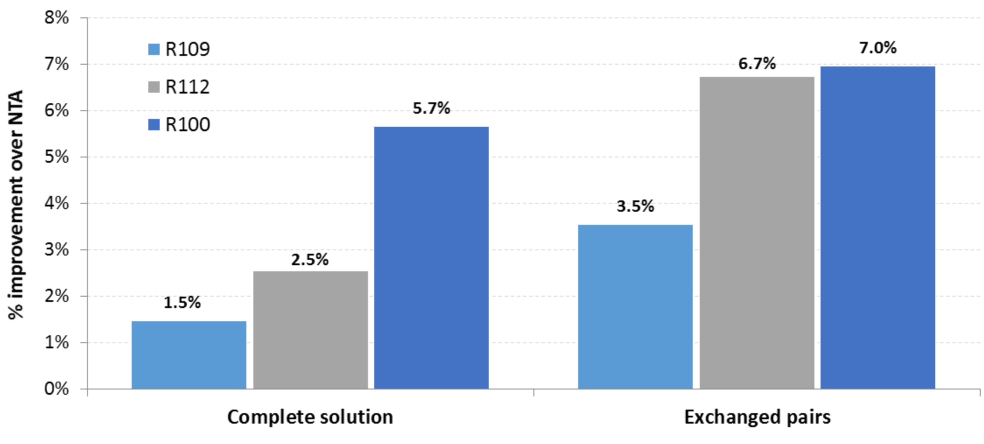
Στο Σχήμα Π.12 αποτυπώνεται η μέση συμπεριφορά των αλγορίθμων LTA (ως ποσοστιαία βελτίωση από τον αλγόριθμο NTA) σε σχέση με τη γεωγραφική κατανομή των απαιτήσεων και την ύπαρξη ή μη χρονικών παραθύρων. Οι αλγόριθμοι LTA φαίνεται μα βελτιώνουν σημαντικά τη λύση σε σχέση με αυτή του NTA σε όλες τις περιπτώσεις. Η βελτίωση τείνει να αυξάνεται στις περιπτώσεις ομαδοποιημένων πελατών (ομάδα C). Επιπρόσθετα, επιτρέποντας μεταφόρτωση σε όλες τις θέσεις των μη εξυπηρετούμενων απαιτήσεων (LTA_d) παρέχονται ιδιαίτερα βελτιωμένα αποτελέσματα συγκριτικά με τη περίπτωση των προκαθορισμένων σημείων (LTA_f).



Σχήμα Π.12. Μέση απόδοση της LTA σε σχέση με τη γεωγραφική κατανομή και χρονικά παράθυρα

<u>Αναδρομολόγηση με μεταφόρτωση για περισσότερα από δύο οχήματα</u>

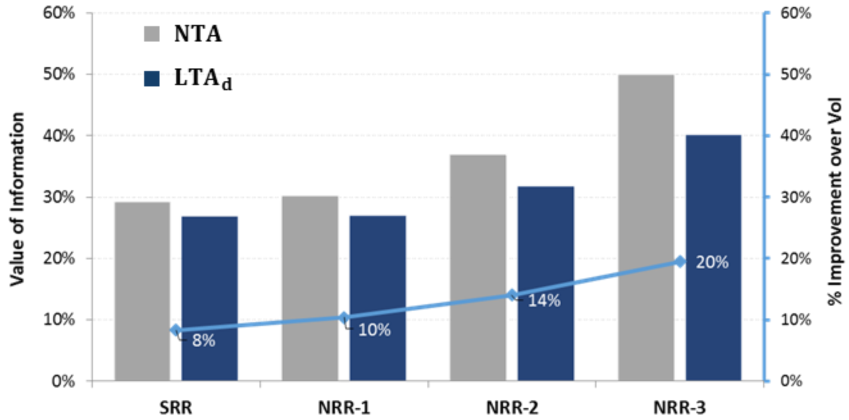
Στην ενότητα αυτή διερευνάται η απόδοση της μεταφόρτωσης σε μία περίοδο αναδρομολόγησης, για πολλαπλά οχήματα. Χρησιμοποιήθηκαν τρία πειράματα R109, R112 και R100, με εύρος παραθύρου 25%, 50% και 100% του T_{max}, αντίστοιχα. Στο Σχήμα Π.13 παρουσιάζονται τα αποτελέσματα των τριών πειραμάτων αναφορικά με τη βελτίωση των λύσεων της LTA συγκριτικά με τις λύσεις της NTA για: α) τη συνολική λύση (όλα τα δρομολόγια), και β) τα ζευγάρια οχημάτων που συμμετείχαν στη μεταφόρτωση. Από αυτό το Σχήμα φαίνεται πως η LTA υπερισχύει και σε αυτή τη περίπτωση της NTA, με μέση βελτίωση 5.7% αναφορικά με τη συνολική λύση. Η βελτίωση φαίνεται να αυξάνεται αναλογικά με την αύξηση του εύρους των χρονικών παραθύρων των πειραμάτων.



Σχήμα Π.13. Μέση απόδοση των LTA για το συνολικό πρόβλημα αναδρομολόγησης

Αξιολόγηση των στρατηγικών αναδρομολόγησης στο ΠΔΔΟΠΜΦ

Τέλος, αξιολογείται η απόδοση των αλγορίθμων LTA συγκριτικά με την απόδοση του αλγόριθμου NTA για το συνολικό δυναμικό πρόβλημα (ΠΔΔΟΠΜΦ – πολλαπλών αναδρομολογήσεων) και με διάφορες πολιτικές αναδρομολόγησης. Για το σενάριο αυτό, χρησιμοποιήθηκε το πείραμα R100 (5 διαφορετικά προβλήματα) και εξετάστηκαν οι πολιτικές SRR, NRR-1, NRR-2 και NRR-3 υπό την τακτική PR. Αναφορικά με την LTA, εξετάστηκε μόνο ο LTA_d αλγόριθμος. Για την αξιολόγηση των αποτελεσμάτων χρησιμοποιήθηκε η μετρική VoI. Στο Σχήμα Π.14 παρουσιάζεται η απόδοση της LTA_d ως ποσοστιαία διαφορά μεταξύ του VoI των δύο αλγορίθμων (LTA_d και NTA) για κάθε πολιτική αναδρομολόγησης. Από το Σχήμα φαίνεται πως και σε αυτή τη περίπτωση η LTA βελτιώνει τις λύσεις που προκύπτουν από την NTA σε κάθε περίπτωση. Η ποσοστιαία βελτίωση αυξάνεται αναλογικά με τη διάρκεια της περιόδου αναδρομολόγησης (συχνότερη αναδρομολόγηση, μικρότερη βελτίωση).



Σχήμα Π.14. Μέση απόδοση της LTA σε σχέση με τις πολιτικές αναδρομολόγησης

ABSTRACT

In this dissertation we studied the Dynamic Vehicle Routing Problem with Mixed Backhauls (DVRPMB), which seeks to assign, in the most efficient way, dynamic pick-up requests that arrive in real-time while a predefined distribution plan is being executed. We used periodic re-optimization to deal with the dynamic arrival of pick-up orders. We developed the formulation of the re-optimization problem, and re-modelled it to a form amenable to applying Branch-and-Price (B&P) for obtaining exact solutions. In order to address challenging cases (e.g. without time windows), we also proposed a novel Column Generation-based insertion heuristic that provides near-optimal solutions in an efficient manner.

Using the aforementioned approach, the dissertation focused on the re-optimization process for addressing the DVRPMB, which comprises a) the re-optimization policy, i.e. when to re-plan, and b) the implementation tactic, i.e. what part of the new plan to communicate to the fleet drivers. We presented and analyzed several re-optimization strategies (combinations of policy and tactic) often met in practice by conducting an extensive series of designed experiments. We did so, by assuming initially unlimited fleet resources under a straightforward objective (i.e. minimize distance traveled). Based on the results obtained, we proposed guidelines for the selection of the appropriate re-optimization strategy with respect to various key problem characteristics (geographical distribution, time windows, degree of dynamism, etc.).

Subsequently, we studied the case in which the number of available vehicles is limited and, consequently, not all orders may be served. To address this, we proposed the required modifications in both the DVRPMB model and the solution approach. By using a conventional objective that strictly maximizes service, we illustrated through appropriate experimentation that the performance of the re-optimization strategies have similar behavior as in the unlimited fleet case. Furthermore, we proposed novel objective functions that account for vehicle productivity during each re-optimization cycle and we illustrated that these objectives may offer improved customer service, especially for cases with relatively high vehicle availability and wide time windows. Moreover, we applied the proposed method to a case study of a next-day courier service provider and illustrated that the method significantly outperforms both current planning practices, as well as a sophisticated insertion-based heuristic.

Finally, we investigated an interesting and novel variant of DVRPMB that allows transfer of delivery orders between vehicles during plan implementation, in order to better utilize fleet capacity and re-distribute its workload as needed in a real-time fashion. We introduced a novel mathematical formulation for the re-optimization problem with load transfers, and proposed an appropriate heuristic that is able to address cases of practical size. We illustrated through extensive experimentation under various operating scenarios that this approach offers significant savings beyond those offered by the previous approaches that do not allow order transfers.

TABLE OF CONTENTS

Chapter 1: Introduction 1

Chapter 2: Literature background 9

2.1 The Vehicle Routing Problem 10
2.2 Related Static Vehicle Routing Problems 13
2.3 Dynamic Vehicle Routing Problems 18
2.4 Branch and Price through Column Generation 29
2.5 Dissertation objectives and contribution 30

Chapter 3: The Dynamic Vehicle Routing Problem with Mixed Backhauls 34

3.1 Problem description 34
3.2 Re-optimization in DVRPMB 36

Chapter 4: Branch-and-Price Algorithm for the Re-Optimization Problem 45

4.1. Overview of the B&P approach 46
4.2. A Set-Partitioning formulation for the proposed Master Problem 48
4.3. The Pricing Sub-Problem 52
4.4. Solution procedure for the pricing sub-problem 56
4.5. The integrated column generation scheme 61
4.6. Solving the integer problem (Branch-and-Price) 63
4.7. A heuristic-based column generation approach 65

Chapter 5: Re-optimization strategies for the case of unlimited vehicle fleet 70

5.1 Re-optimization and the routing environment 71
5.2 Re-optimization strategies 74
5.3 Theoretical insights for re-optimization strategies 76
5.4 Computational experiments 80
5.5 Concluding remarks 92

Chapter 6: The DVRPMB for the case of limited resources 94

6.1 Introduction and background 95
6.2 Objective functions for the *m*-DVRPMB 97
6.3 Modifications in the B&P algorithm to deal with limited resources 109
6.4 Computational experiments 112
6.5 Case study in a Courier environment 123

Chapter 7: The DVRPMB with Load Transfers 134

7.1 Introduction...134
7.2 Related literature on transfer (transshipment) operations 136
7.3 Re-optimization in DVRPMB-LT 139
7.4 Solution approach for DVRPMB-LT($\boldsymbol{\ell}$) 150
7.5 Computational experiments 157

Chapter 8: Conclusions and future research 170

8.1 Conclusions...170
8.2 Future research...173

References 176

Appendices 194

Appendix A. Detailed results of the experiments presented in Chapter 5 194
Appendix B. Detailed results of the experiments presented in Chapter 6 201

LIST OF FIGURES

Figure 1.1. Example of the re-optimization problem 3
Figure 2.1. From MTSP to DVRPMB 10
Figure 2.2. A solution example of the VRP 12
Figure 3.1. Overview of the re-optimization framework........ 38
Figure 3.2. A simple re-optimization process 39
Figure 3.3. Sets of vehicles and orders considered in the static problem........ 40
Figure 3.4. Computational time increases prohibitively as number of requests increase........ 44
Figure 4.1. Illustration of the decomposition approach for the pricing sub-problem........ 53
Figure 4.2. Sequential and parallel implementations of the Pricing Sub-problem........ 61
Figure 4.3. The column generation procedure at each re-optimization cycle 62
Figure 4.4. Example of the column generation process 63
Figure 4.5. Pseudo-code of heuristic approach for generating columns for vehicles *en route*........ 68
Figure 5.1. Example in which (a) re-optimizing upon the receipt of each DO yields an inferior result than (b) re-optimizing when both DO are received........ 72
Figure 5.2. Example in which (a) re-optimizing when both DO are received yields an inferior result than (b) re-optimizing upon the receipt of each DO 73
Figure 5.3. Illustration of re-optimization policies (NRR is applied for N=3; FTR is applied every hour) 74
Figure 5.4. Full release (FR) vs. partial release (PR) tactic for a single re-optimization cycle and a single route........ 75
Figure 5.5. Planned and actual routes under both tactics for three re-optimization cycles (solid line is the planned route, dotted line the executed route)........ 79
Figure 5.6. Geographical distribution patterns of Solomon benchmarks; (a) Uniform distribution (R1 and R2), (b) Clustered case (C1 and C2), (c) Semi-clustered case (RC1 and RC2); blue circles represent the customer locations and red square the depot. 81
Figure 5.7. Time window information for the Solomon datasets (bars indicate the range between the min & max TW values of each instance; squares represent the average TW width) 82
Figure 5.8. (a) Number of DO per hour for a 3-day period; (b) average number of DO per hour (received and served) 83
Figure 5.9. Average performance of re-optimization strategies for the different datasets........ 86
Figure 5.10. Average performance of re-optimization strategies for different TW pattern groups 88
Figure 5.11. Average performance of re-optimization strategies for different $edod$ levels 89
Figure 5.12. Performance of policies under the PR tactic for various geographical distributions and dod levels........ 90
Figure 5.13. Performance of policies under FR tactic for various geographical distributions and dod levels........ 90
Figure 5.14. Comparison of the effects OPT and HEUR........ 91

Figure 6.1. The profit per order according to the three objective functions employed 98
Figure 6.2. Example for comparing between alternative expressions of objective $z2$ 102
Figure 6.3. Customer topology for Example 1 of Statement 2 105
Figure 6.4. Example 1 for Statement 2 (blue color corresponds to vehicle $K1$ and red to $K2$) 106
Figure 6.5. Network representation for Example 2 of Statement 2 107
Figure 6.6. Solution provided under both objectives for Example 2 of Statement 2 108
Figure 6.7. Average percentage of DO served vs. the available vehicles at depot per dataset............ 114
Figure 6.8. Average performance of re-optimization strategies for the different datasets 117
Figure 6.9. Performance of strategies for R1 dataset for various values of fleet availability 117
Figure 6.10. Performance of strategies for C1 dataset for various values of fleet availability 118
Figure 6.11. Overall average performance of objectives per investigated instance 119
Figure 6.12. Average performance of objectives w.r.t. re-optimization policy and tactic 120
Figure 6.13. Average performance of objectives for different TW pattern groups and vehicle availability values 121
Figure 6.14. Average number of served DO per objective w.r.t. re-opt. frequency and tactic............ 122
Figure 6.15. A typical model of courier service operations 124
Figure 6.16. Typical routing process (the initial routing plan is generated at t=0) 124
Figure 6.17. Representation of customer locations for the busiest day on a digital map (blue circles correspond to static customers and red crosses to DO) 126
Figure 6.18. Distribution of DO received w.r.t. the time of day (all three days) 126
Figure 6.19. Arrival and TW patterns of DO for Day 3 (a blue star reflects the time each DO was received; the bars indicate the related TW) 127
Figure 6.20. Example of considering combinations in the local update heuristic 129
Figure 6.21. Overall performance of all scenarios (average of all days).. 132
Figure 6.22. Overall performance of scenarios w.r.t. the working hours per vehicle 132
Figure 6.23. Average performance of scenarios w.r.t. unit cost (cost per request) 133
Figure 7.1. Example of (a) overlapping, and (b) non-overlapping vehicle routes.............................. 135
Figure 7.2. The re-optimization process.. 139
Figure 7.3. Illustrative example for the DVRPMB-LT(ℓ); (a) routes prior to re-optimization, (b) solution of DVRPMB without transfers, (c) load transfers at fixed location; (d) load transfers at customer location 142
Figure 7.4. Representation of the transfer location 143
Figure 7.5. Modeling the case of transferring loads at customer location i 143
Figure 7.6. Allowable arcs $(i, j) \in A$ (possible connections from any i to any j, $\forall i, j \in W$) 145
Figure 7.7. Solution framework for DVRPMB-LTℓ (LTA) 151
Figure 7.8. Pseudo-code for identifying the most appropriate transfer location among the locations of the customers not yet served 155
Figure 7.9. Pseudo-code of heuristic approach for incorporating the fixed transfer location 156
Figure 7.10. Interchange moves; (a) within a single route, (b) between a route pair 157

Figure 7.11. Overall heuristic assessment (deviation from optimal) 161
Figure 7.12. Average performance of LTA w.r.t. the number of delivery orders per vehicle 162
Figure 7.13. Average performance of LTA w.r.t. geographical distribution and TW patterns 165
Figure 7.14. Average performance of LTA on the full re-optimization problem 168

LIST OF TABLES

Table 2.1. VRP taxonomy w.r.t. information evolution and quality 13
Table 2.2. Types of solution methods for the VRPTW 14
Table 2.3. Types of solution methods for the VRPB 16
Table 4.1. Acceleration techniques for the solution of ESPPTWCC for the DVRPMB 59
Table 5.1. Customer coordinates for counter-example of Claim 2 78
Table 5.2. Routing costs under both tactics for three re-optimization cycles 78
Table 5.3. Performance of heuristic B&P algorithm 84
Table 5.4. Average computational effort of re-optimization strategies per dataset for 50% dod 86
Table 5.5. Classification of instances in TW-pattern groups 87
Table 5.6. Classification of instances in different $edod$ levels 88
Table 5.7. Preferable re-optimization policies per parameter 93
Table 6.1. Test instances 113
Table 6.2. Performance of heuristic B&P algorithm 115
Table 6.3. Classification of investigated instances in TW-pattern groups 120
Table 6.4. Key indicators of the case study (number of customers and resources) 125
Table 6.5. Average number of customers served per vans and scooters 127
Table 6.6. Components involved in the experimental investigation 130
Table 6.7. Planning scenarios 131
Table 6.8. Total routing results (in hours) for all scenarios on the 3-day period 131
Table 7.1. Key information in references investigating transshipment operations 138
Table 7.2. Notation for DVRPMB-LT(ℓ) 144
Table 7.3. Parameters of test problems 158
Table 7.4. Description of the algorithms employed 159
Table 7.5. Detailed results for NTA and the investigated LTA algorithms 159
Table 7.6. Test cases 163
Table 7.7. Customer characteristics of the generated instances 164
Table 7.8. Detailed results for NTA and the proposed LTA algorithms 164
Table 7.9. Information of each investigated instance during the re-optimization cycle 166
Table 7.10. Detailed results for NTA and LTAd 167
Table 7.11. Performance of LTA and NTA per re-optimization policy 169

LIST OF ABBREVIATIONS

Abbreviation	Description
1-1 PDP	One-to-one Pickup and Delivery Problem
1-M-1 PDP	One-to-many-to-one Pickup and Delivery Problem
ALNS	Adaptive Large Neighborhood Search
B&B	Branch-and-Bound
B&P	Branch-and-Price
CG	Column Generation
DARP	Dial-A-Ride Problem
DARPT	Dial-A-Ride Problem with Transfers
DO	Dynamic Orders; requests that are received in real-time, i.e. during execution
DoD	Degree of Dynamism
DPDP	Dynamic Pickup and Delivery Problem
DVRP	Dynamic Vehicle Routing Problem
DVRPMB	Dynamic Vehicle Routing Problem with Mixed Backhauls
DVRPMB(ℓ)	The re-optimization problem of DVPMB
DVRPMB-LT	Dynamic Vehicle Routing Problem with Mixed Backhauls and Load Transfers
DVRPMB-LT(ℓ)	The re-optimization problem of DVPMB-LT
DVRPMBTW	Dynamic Vehicle Routing Problem with Mixed Backhauls and Time Windows
ESPPTWCC	Elementary Shortest Path Problem with Time Windows and Capacity Constraints
FR	Full Release tactic
FTR	Fixed-Time Re-optimization
GA	Genetic Algorithm
GPS	Global Positioning System
GRASP	Greedy Randomized Adaptive Search Procedure
GUB	Global Upper Bound
HEUR	The heuristic Branch-and-Price approach
HVRP	Heterogeneous Vehicle Routing Problem
LB	Lower Bound
LDS	Limited Discrepancy Search
LP-SPP	Linear relaxation of Set-Partitioning Problem
LTA	Load Transfer Algorithm
LTA^p	Load transfer algorithm for a single pair of vehicles
m-DVRPMB	Dynamic Vehicle Routing Problem with Mixed Backhauls and Limited Resources
MILP	Mixed Integer Linear Programming

Abbreviation	Description
M-M PDP	Many-to-many Pickup and Delivery Problem
MP	Master Problem
NRR	Number of Requests Re-optimization
NTA	No Transfer Strategy
OPT	The Exact Branch-and-Price approach
P&D	Pickup and delivery operations associated with the same customer
PDP	Pickup and Delivery Problem
PDPT	Pickup and Delivery Problem with Transfers
PDPTWT	Pickup and Delivery Problem with Time Windows and Transfers
PR	Partial Release tactic
RMP	Restricted Master Problem
RT-VRPTWDP	Real-Time Vehicle Routing Problem with Time Windows and simultaneous pick-up/delivery demands
SO	Static Orders; offline requests assigned to vehicles prior to start of operations
SPP	Set-Partitioning Problem
SRR	Single-Request Re-optimization
TSP	Traveling Salesman Problem
TW	Time Window
VoI	Value of Information
VRP	Vehicle Routing Problem
VRPB	Vehicle Routing Problem with Backhauls
VRPCB	Vehicle Routing Problem with Clustered Backhauls
VRPDSPTW	Vehicle Routing Problem with Deliveries, Selective Pickups and Time Windows
VRPMB	Vehicle Routing Problem with Mixed Backhauls
VRPMBTW	Vehicle Routing Problem with Mixed Backhauls and Time Windows
VRPMBTW	Vehicle Routing Problem with Clustered Backhauls and Time Windows
VRPTW	Vehicle Routing Problem with Time Windows

GLOSSARY OF SYMBOLS

Notation	Description
0	Depot location
$0'$	Transfer location corresponding to the depot
A	Set of arcs connecting all nodes in W
a_i	The opening time of the time window of order i, $\forall i \in N$
b_i	The closing time of the time window of order i, $\forall i \in N$
c	Cost notation
C	Subset of static or committed orders during re-optimization
C_k	Set of assigned static orders to be serviced by each vehicle *en route* $k \in K$
c_{ij}	The cost of traversing arc $(i, j), \{i \in W, j \in W \backslash M\}$
c'_{ij}	Modified cost associated with arc $(i, j) \in A$; i.e.
$\tilde{c}_{\delta i}$	The accumulated cost of partial path δ ending to vertex $i \in N$
$\bar{\bar{c}}_{\delta i}$	A cost factor (equilibrium cost) representing an upper bound of the total modified cost required to serve all committed orders not yet included in partial path δ
d_i	The demand/supply of the order at each client site
e_{ir}	Binary coefficient that equals 1 if order $i \in N$ is included in route $r \in \Omega$.
F	Subset of dynamic or flexible orders during re-optimization
$f(u)$	Finish node of transfer location $u \in U$
$f(U)$	Set of all finish nodes of transfer locations, i.e. $f(U) = \{f(u): u \in U\}$
G	The complete direct graph on Euclidean plane
h_i	The arrival time of a new order, $\forall i \in C \cup F$
K	Set of vehicles *en route* (finite number)
K_d	Set of available vehicles located at the depot
ℓ	Re-optimization cycle notation
L	Number of re-optimization cycles
M	Set of starting location of vehicles, i.e. $M = \cup_{k \in K}\{\mu_k\}$
M'	Set of transfer locations corresponding to starting location of vehicles
n	Number of static (delivery) orders
$\tilde{n}$	Number of dynamic (pick-up) orders
N	Set of customer nodes (orders), $N = C \cup F$
N'	Set of transfer locations corresponding to customer nodes
$O(\delta i)$	Set of committed orders included in partial path δ ending at vertex i
$O'(\delta i)$	Set of all remaining orders $N = C \cup F$ not yet served by partial path δ
$\bar{Q}$	The available capacity of any vehicle $k \in V$

Notation	**Description**
Q_{ik}	Load variables provide the load of vehicle $k \in V$ immediately after the service at node $i \in W$
$\tilde{q}_k$	The load of pick-up orders currently loaded in the vehicle during previous re-optimization cycles, $\forall k \in K$
r	Notation for a route-column in column-generation algorithm
R_s	Set of initial routes (prior to start of execution)
s_i	The order's service time, $\forall i \in N$
$s(u)$	Start node of transfer location $u \in U$
$s(U)$	Set of all start nodes of transfer locations, i.e. $s(U) = \{s(u): u \in U\}$
S_ℓ	Solution of re-optimization cycle
t_{ij}	The travel time between nodes $(i, j), \{i \in W, j \in W \backslash M\}$
T_ℓ	Re-optimization time instance, $\ell = 1, \ldots, L$
T_{max}	Available time horizon
u	Transfer location notation
U	Set of all transfer location nodes, i.e. $U = U_f \cup \{0\}' \cup M' \cup N'$
U_f	Fixed transfer location node(s)
V	Set of all vehicles
W	Set of all nodes involved, i.e. $W = N \cup M \cup \{0\} \cup U$ (where $U \neq \emptyset$ only for DVRPMB-LT)
w_{ik}	Time variables specifying the time vehicle $k \in V$ starts serving customer $i \in N$
x_{ijk}	Binary flow variables equal to 1 if arc $(i, j) \in A$ is used by vehicle $k \in V$
y_r	Binary coefficient which equals to 1 if route $r \in \Omega$ is used
Z	A very large positive constant
z_j^{ki}	Binary variables equal to 1 if order $i \in N$ is onboard vehicle $k \in K$ when it arrives to node $j \in W \backslash M$, and 0 otherwise, for all $i \in N, k \in K$
Γ_ℓ	The (static) re-optimization problem solved
$\Lambda_{\delta i}$	Label represented by a vector related to partial path δ ending to vertex $i \in N$
μ_k	Current position of each vehicle $k \in K$
$\tilde{\Xi}$	Constant value to simulate the time needed for the load to remain at the transfer location (till its departure).
ξ_u	A fixed profit assigned to each dynamic order served
ξ_p	A profit in case an order is served within the upcoming re-optimization cycle
ξ_c	A penalty related to the routing costs
π_i^u	The value of the dual variable in the dual solution of RMP at iteration u of the CG algorithm
τ	Predefined time interval (e.g. the last hour of the available working period)
Φ	Penalty assigned for each dynamic order not served
Ψ_k	Independent sub-problem, $k = 1, \ldots, K + 1$
Ω	Set of all feasible routes (columns) in the set-partitioning formulation, i.e. $\Omega = (\bigcup_{k \in K} \Omega_k) \cup \Omega_p$
Ω_k	Columns correspond to vehicles K already *en route*
Ω_p	Columns correspond to vehicles K_d located at the depot

Chapter 1: INTRODUCTION

Delivery and collection of goods accounts for a significant part of supply chain costs. Therefore, planning the distribution and pick-up of goods in an efficient manner is an appropriate way to reduce logistics costs, while, at the same time, improve the quality of service. In the attempt to address these issues, significant research has been conducted in vehicle routing. The majority of this research has focused on deterministic and static models, in which all information and problem parameters are assumed to be known in advance, and the related decisions are made prior to the start of plan execution.

In practice, however, many factors may cause disruptions in the execution of the original distribution (and/or pick-up) plan. These usually stem from the occurrence of dynamic events, such as delays due to traffic congestion, unavailability of docking space, vehicle breakdowns, temporary alterations in the road network, etc. Moreover, increasing competitive pressures and expectations for high-quality service have led urban logistics operators to enhance their offering by responding to requests that arrive in a dynamic fashion. For example, dynamic arrival of orders, while the delivery and/or pick-up plan is being executed, is common in many practical applications, including courier, money-transfer and repair-maintenance services. In all these applications, oftentimes, only a moderate portion of the requests for service (orders) are known in advance, and there exists an initial routing plan (*a priori* plan) that assigns those known requests to the available fleet. The dynamic orders, which arrive during plan execution, must be assigned to appropriate vehicles in real-time. Incorporating dynamic orders in the *a priori* plan may reduce the plan's quality or, even worse, may lead to infeasibilities.

Real-time decision-making appears to be an effective option for addressing dynamic situations[2]. In this approach, the *a priori* plan can be modified and updated (once or repeatedly) based on the real-time state of the logistics system. The updating process is also made possible and practical by the latest advances in fleet telematics, which are capable of: a) providing dispatchers with real-time information on the status and location of vehicles, status of customers, as well as network conditions, and b) transmitting the related re-optimization decisions to the fleet drivers in an effective manner.

The process of updating the plan in real-time to incorporate the dynamic orders needs to consider two fundamental issues: a) The problem of re-optimizing (re-planning) the vehicle routes at a certain time instance considering the unserved customer orders (and their *a priori* assignment to vehicles), as well as all dynamic orders known up to that time instance, and b) the re-optimizing (re-planning) process; i.e. when to re-plan and which dynamic orders to release to the vehicles for execution. This dissertation focuses on both these issues.

Re-optimization problem and re-optimization process

Figure 1.1 provides an illustrative example for the **re-optimization problem**. Consider a fleet of vehicles initially located at a single depot ($t = 0$). The fleet is homogeneous and each vehicle is assigned a route (Figure 1.1a). Customer orders arrive over time ($t > 0$) during the execution of the initial routes through a call center. At time instance t_1, the dispatcher decides to incorporate the newly arrived (dynamic) orders in the partially executed plan (Figure 1.1b). Note that at time t_1, some customers have been already served and the available resources (e.g. fleet vehicles and their capacities, available time horizon, etc.) are limited. The dispatcher assigns some, or all, new orders to the available vehicles *en route* or to vehicles located at the depot, if any. Also the dispatcher may also reassign orders between the available vehicles, if this is possible. It is clear that the decisions to be made are often complex, since many assignment options are available. For example, as shown in Figure 1.1c, order d may be assigned to one of two routes. Note also that due to problem constraints, customers may even be denied service. Thus, the underlying objective is to serve all static orders and as many as possible (or ideally all) dynamic orders, respecting all constraints (e.g. vehicle capacity, shift remaining time, etc.), while minimizing a cost metric for the entire fleet (Figure 1.1d).

[2] Note that there are other options for dealing with such problems. For example, probabilistic information for future events can be incorporated and taken into account during (re)optimization (Powel, 1996).

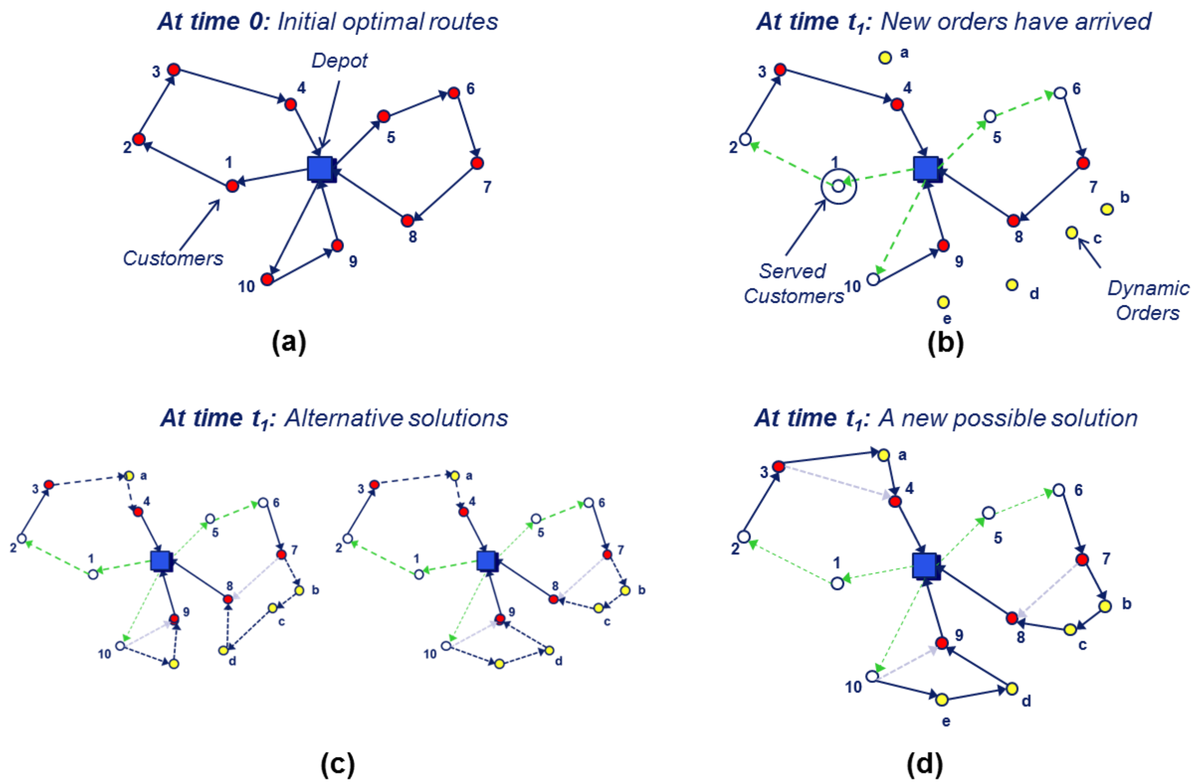


Figure 1.1. Example of the re-optimization problem

Based on the above illustrative example, we can define the Re-optimization Problem, as follows:

Description 1.1.: The Re-optimization Problem

Consider a fleet of vehicles executing a certain delivery/pick-up plan. Consider also that new requests for service (orders) arrive dynamically during execution. The problem seeks to assign as many of the new orders as possible to vehicles or/and reassign orders between the available vehicles, if possible, while achieving efficient routing cost, and respecting all service constraints.

The re-optimization problem comprises a single step of the **re-optimization process**. A typical re-optimization process comprises the following decisions and actions:

- Select a sequence of re-optimization periods (re-optimization cycles), not necessarily of equal duration. The dynamic orders arriving within each period (and perhaps some orders of the previous periods not yet served) are planned for execution by the fleet at the end of the period. The selection of an appropriate sequence of re-optimization periods is a challenging issue, which depends on several aspects, including the rate of the arriving orders, the characteristics of the plan been executed, etc.

- At the end of each re-optimization period, solve the re-optimization problem described above.
- Communicate part of, or the entire new plan, to drivers and repeat in every re-optimization cycle.

In addressing the re-optimization problem and the re-optimization process, one needs to consider significant operational characteristics, which are relevant virtually in every dynamic vehicle routing problem (see Table 1.1).

Table 1.1. Operational characteristics of re-optimization in dynamic vehicle routing

Category	Characteristic/Description
Time	*Planning horizon* The planning horizon refers to the available working period of the vehicles to serve customers (e.g. driver's shift). Various planning horizon options such as flexible shifts, rolling shifts, etc. may offer different advantages. *Arrival pattern of dynamic orders* The timing of order arrival may play a significant role in the re-optimization process. For example, an arrival pattern with significant peaks (i.e. dynamic orders concentrated around certain times) may require a different sequence of re-optimization periods than a case in which dynamic orders arrive more or less uniformly over time. *Re-optimization cycle* Long re-optimization intervals limit the dispatcher's options (since a larger portion of the route has been completed) and may lead to lost opportunities regarding favorable insertion locations for the newly arrived orders. On the other hand, short re-optimization intervals (frequent re-optimization) may not consider adequately favorable dependencies between arriving orders (adequately rich order combinations). Note also that in practice, frequent changes of the delivery plan may cause nervousness to the system. *Time windows* Time windows, which refer to the interval within which each customer may be served, may lead to considerable "dead" times due to vehicles waiting for a customer's time window to open. "Dead" times may be better exploited in cases of frequent re-optimization, since frequent re- planning may use vehicle waiting times for servicing a nearby dynamic order.

Space	The spatial distribution of customers is another essential parameter in both static and dynamic vehicle routing problems. For example, in the case in which customers form distinct groups (clustered case), the excess cost of an additional visit within the same cluster tends to be low, while an inter-cluster visit is expensive. If the customers are uniformly distributed in space, then the excess cost of a visit may vary widely.
Distribution environment and practice	*Distribution environment* The characteristics of the distribution environment may also play an important role in the dispatcher's decisions. Such characteristics may include a) traffic congestion (high or low) and b) the type of service (pick-up, delivery or mixed). For example, highly congested areas decrease the service rate and may limit the options of re-optimization. Furthermore, in case of delivery only operations, orders may be serviced by the vehicle initially assigned to them, limiting the re-optimization options. To overcome this limitation, order exchanges between vehicles is necessary, increasing the complexity of the distribution process. *Distribution practice* In this context, distribution practice refers to the way resources are deployed to handle dynamic orders. For example, should all available vehicles be dispatched, or should some vehicles be kept at the depot to only serve dynamic orders? Additional examples include positioning of vehicles at forward points in anticipation of (dynamic) order assignments, or transfer loads (exchange) among vehicles as necessary to streamline routing operations.

Dissertation motivation and focus

The research in this dissertation has been motivated by practical courier applications (Ninikas *et al.*, 2014). For example, in a typical courier setting, a fleet of delivery vehicles originating from a local distribution hub (depot) is tasked to deliver or pick-up orders known prior to the start of operations (static orders). As the work plan unfolds, however, customer orders are received through a call center, for on-site pick-up within the current period of operations. These pick-up orders have to be collected and returned to the hub for further processing. In this work, we seek to allocate in real-time dynamically arriving (pick-up) orders to the most appropriate vehicles, either to those *en route* or to extra vehicles stationed at the depot.

Beyond the courier case, such problems arise naturally in money-transfer and repair-maintenance services. Service vehicles are called to serve requests for money pick-ups or faulty equipment repairs, respectively, which arrive to a dispatch center in a dynamic fashion. Related examples may also be found in coach transfers, in which vehicles that execute planned routes originating from major locations (e.g. airport) and serving predefined drop-off areas (e.g. accommodation sites), are requested to collect passengers from additional locations while *en route*.

The problem investigated in this dissertation comprises a dynamic version of the *one-to-many-to-one* pick-up and delivery problems (1-M-1-PDPs, Berbeglia *et al.*, 2008; 2010; Gribkovskaia and Laporte, 2008). The term "one-to-many-to-one" denotes that vehicles deliver commodities initially loaded at the depot to customers (linehaul customers), while other commodities are picked up from customers and are transported back to depot (backhaul customers). Our case considers that a) each customer requires only pick-up or delivery, and b) pick-up and delivery customers may be served in an arbitrary order. The static version of this problem can be found in the literature as the Vehicle Routing Problem with Mixed Backhauls and Time Windows (VRPMBTW) as introduced by Kontoravdis and Bard (1995). For that reason, we refer to our problem as the *Dynamic Vehicle Routing Problem with Mixed Backhauls (DVRPMB)*. To the best of our knowledge, the dynamic version of 1-M-1 PDPs and especially the DVRPMB has yet to be investigated (Parragh *et al.*, 2008).

In this dissertation, we approach DVRPMB by solving repeatedly static re-optimization problems. For the latter, we define the **re-optimization model and propose a Branch-and-Price (B&P) approach** to obtain exact solutions. In order to address challenging cases (e.g. without time windows), we propose a novel Column Generation-based insertion heuristic that provides near-optimal solutions in an efficient manner.

Using the aforementioned fundamental approach, the dissertation drills down to the **re-optimization process** for addressing the DVRPMB. As mentioned above, the problem environment plays a significant role in this process. The rate of delivery, the rate of arrival of new orders, the space and temporal distribution of the orders, and the percentage of dynamic orders are some of the operational characteristics that may affect the adoption of the appropriate re-optimization process. In this dissertation, we consider various problem settings in order to provide basic guidelines in terms of managing various dynamic scenarios in a flexible and cost effective manner. Considering these operational characteristics, we propose and analyze several re-optimization policies often met in practice by conducting an extensive series of designed experiments. In addition, we investigate those policies in combination with different tactics regarding the part of the plan that is released for implementation.

Initially, we focus our study on the case of unlimited fleet using a straightforward objective (i.e. minimize distance traveled). Subsequently, we examine the case of DVRPMB, in which the **number of available vehicles is limited**. We introduce appropriate objective functions that account for both the service provided (in terms of orders served), and the cost of service.

Finally, we examine the possibility of relaxing the intrinsic constraint of preventing delivery orders to be reassigned to other vehicles, which may impose significant limitations to re-optimization and may lead to inability of servicing some newly received orders. Thus, we study the re-optimization problem by **allowing delivery orders to be transferred between vehicles** during the execution of the plan. By doing so, we attempt to better utilize the fleet by re-distributing its workload as needed in a real-time fashion.

For the above problems cases, we have accomplished the following:

- Proposed an appropriate periodic re-optimization process for the DVRPMB
- Proposed an exact and a novel heuristic approach in order to solve the underlying re-optimization problem of DVRPMB
- For the full dynamic case, presented and analyzed a) re-optimization tactics regarding the implementation of the plan, and b) re-optimization policies regarding the re-optimization frequency and the resulting solution quality. Proposed guidelines on re-optimization depending on the characteristics of the dynamic environment
- Proposed novel objective functions to address the case of limited fleet in DVRPMB that account for vehicle productivity, and investigated the effectiveness of these objective functions on the quality of the solutions for various characteristics of the dynamic routing environment
- Validated the practicality of the proposed methods through a large industrial case of a next-day courier service provider
- Introduced the case of load-transfer operations during the execution of the routing plan, and proposed a novel mathematical model for the underlying re-optimization problem.
- Developed a new heuristic method to solve this problem and compared the results obtained with operations that do not allow load-transfers.

The remainder of the dissertation is organized as follows:

Chapter 2 presents and discusses the related problems in the literature, the most significant approaches used, and the similarities and differences with respect to the problem studied in this dissertation. Chapter 2 also identifies the related research gaps, as well as the contributions of the dissertation.

Chapter 3 presents a formal description of the problem in hand. This is followed by the model of the (static) re-optimization problem considered in each re-optimization cycle. An overview of the solution framework is also given.

Chapter 4 presents the Branch & Price (B&P) approach proposed to solve the re-optimization problem (of DVRPMB). This approach includes restructuring the problem to be amenable to column generation (CG), as well as required modifications to the conventional B&P approach so that it applies to the problem in hand. A novel CG-based insertion heuristic is also proposed to provide near optimal solutions in an efficient manner for computationally demanding cases (e.g. without time windows).

Chapter 5 studies the re-optimization process for the case of unlimited fleet. Several re-optimization strategies are discussed and analyzed. Based on the results obtained we propose re-optimization guidelines under various operational settings.

Chapter 6 deals with the case of limited fleet. We describe the required modifications to the approach developed for the unlimited fleet case, and introduce appropriate objective functions. Moreover, we apply our proposed method to a real case of a next-day courier service provider.

Chapter 7 introduces and examines a variant of the re-optimization problem that allows orders to be transferred between vehicles during execution. The Chapter provides an arc-based formulation for the re-optimization problem and an appropriate heuristic that is able to address (solve) cases of practical size. This approach is compared to the previous ones that do not allow transfers.

Finally, Chapter 8 presents the conclusions of this dissertation, the theoretical and practical contributions, along with directions for further research.

Chapter 2: LITERATURE BACKGROUND

As already mentioned in Chapter 1, this dissertation focuses on dynamic routing problems. The latter concern the dynamic version of Vehicle Routing Problems (VRPs), including the VRP with Backhauls and the Pickup and Delivery Problem (PDP). The VRP is essentially a Multiple Traveling Salesman Problem (MTSP) with a capacity constraint for each salesman. Likewise, other constraints and assumptions can be added to the basic form of VRP in order to take into account key aspects of distribution and scheduling, resulting into different VRP variants.

Figure 2.1 illustrates how the Dynamic Vehicle Routing Problem with Mixed Backhauls (DVRPMB), which is the problem addressed in this dissertation, can be derived from MTSP by adding appropriate constraints. In the following Sections, the VRP with Time Windows (VRPTW), the VRP with Backhauls (VRPB), the Pickup and Delivery Problem (PDP) and the Dynamic VRP (DVRP) are reviewed. All are related to DVRPMB.

Specifically, Section 2.2 overviews the static version of the problems related to DVRPMB, i.e. VRPTW, VRPB and PDP. Section 2.3 discusses the dynamic versions of VRP and PDP and related solution strategies to address dynamism. Section 2.4 provides a targeted discussion on the essentials of the basic technique employed in this dissertation (Branch-and-Price), and Section 2.5 highlights the contributions of the dissertation.

Aspects of the literature that are quite specific to particular topics of the dissertation are presented and discussed within the corresponding Chapters.

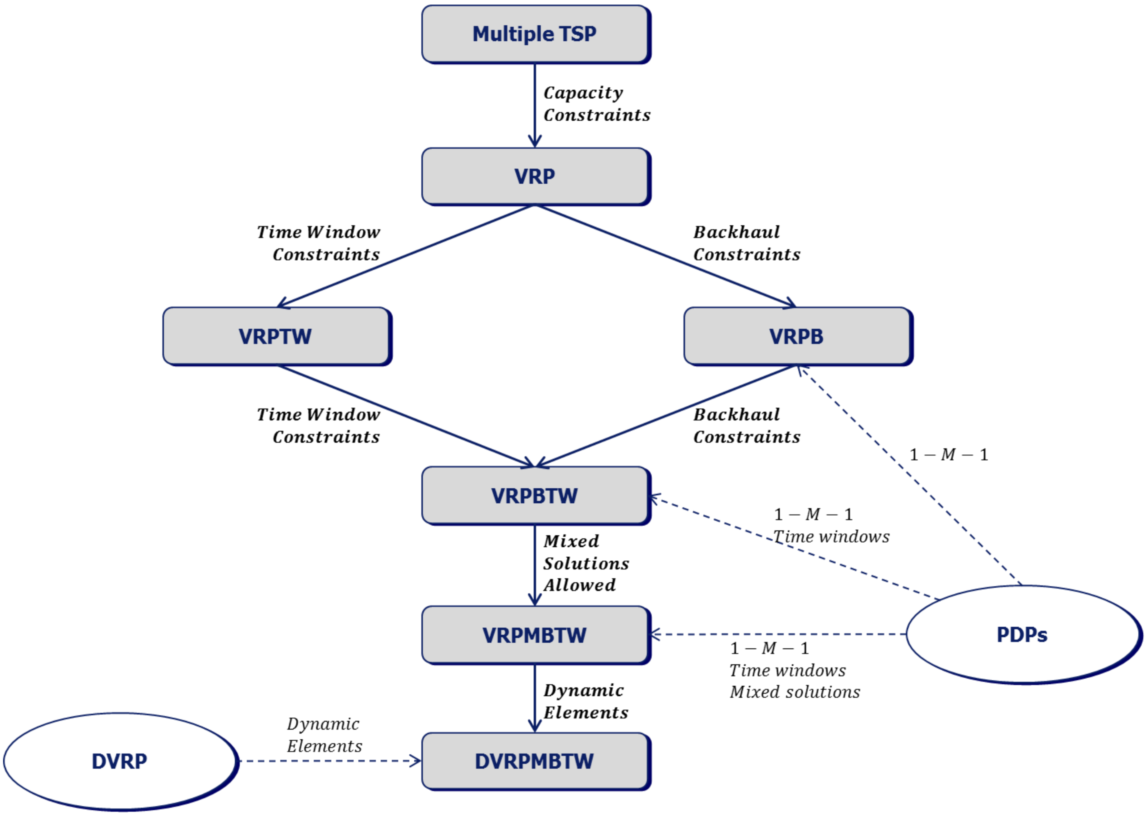


Figure 2.1. From MTSP to DVRPMB

2.1 The Vehicle Routing Problem

The vehicle routing problem (VRP) is one of the most studied problems in the field of Operations Research and many mathematical programming techniques have been developed to solve it. VRP applications are of critical importance to aspects of logistics management, since they provide decision support to complex practical transportation and distribution problems. Efficient decisions on related applications may have significant impact on operating costs. Numerous practical applications have illustrated that the use of computerized procedures for planning the distribution process result in substantial savings (generally from 5% to 20%) in transportation costs (Toth and 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 consists of designing the optimal set of routes for a fleet of vehicles in order to serve a certain set of customers. It was firstly introduced by Dantzig and Ramser (1959), who proposed a mathematical programming formulation and an algorithmic approach to solve a practical problem of delivering gasoline to service stations. The definition of VRP and its variants, as well as an extensive analysis of solution methods, are presented by Toth and Vigo (2002). Currently numerous commercial software applications are available that embed advanced algorithmic approaches for solving different practical VRP cases.

In a typical VRP setting, customers are represented by nodes of a network, they have known demand, and each must be served once by only one vehicle. Every arc (i, j) of the network (where i and j are network nodes) is associated with a cost c_{ij} representing the cost of traveling from i to j. Each vehicle has a certain capacity and its route must start and end at a certain depot. The total demand of those customers served by a vehicle may not exceed the vehicle's capacity. The objective of the problem is to minimize the total cost traveled by all vehicles. Figure 2.2 illustrates a feasible solution of a VRP for a given set of customers.

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

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

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

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 of the corresponding m-traveling salesman problem (m-TSP)

μ_i = the demand at location i

Q = the vehicle capacity

The VRP is modeled as an integer-programming problem and corresponds to an NP-hard problem (Lenstra and Kan, 1981); therefore, practical (large) problem instances cannot be solved to optimality within reasonable time. As a result, exact solution methods are used for limited-size problem instances, while heuristics and metaheuristics are normally applied for practical cases. There is extensive literature regarding methods for solving the VRP. The interested reader may refer to the work of Toth and Vigo (2002), Christofides *et al.* (1979), Desrochers *et al.* (1990), Laporte (1992), Golden and Assad (1988), and Laporte and Osman (1995).

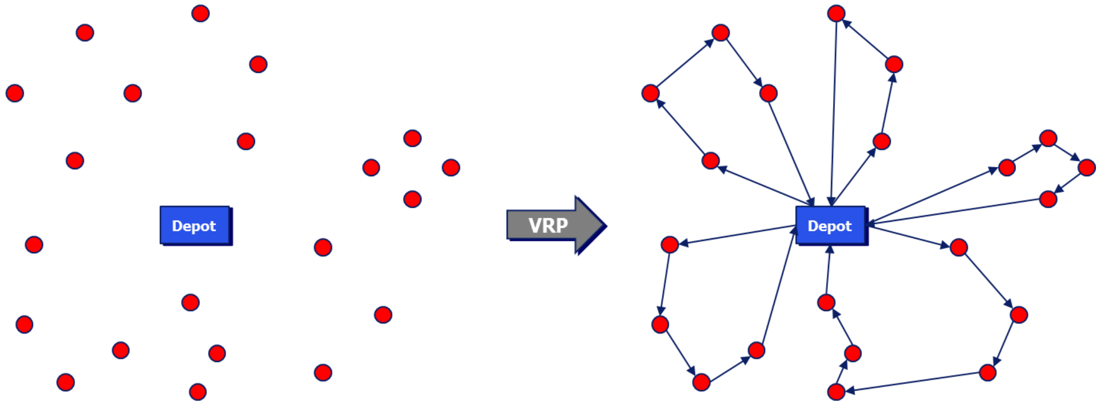


Figure 2.2. A solution example of the VRP

Beyond the classical VRP formulation, a number of problem variants have been studied, depending on the constraints of the problem in hand. Among the most common are the VRP with Time Windows (VRPTW), in which each customer must be visited within a certain, predefined time interval; the VRP with Pickup and Delivery (PDP), in which each customer is associated with two service locations, one for the pick-up and another for the delivery of goods; the Heterogeneous fleet VRP (HVRP), in which involved vehicles have different capacities; and the VRP with Backhauls (VRPB), in which a set of customers require the delivery of goods to their locations, while another set requires picking up the goods from their locations and returning them back to the depot. VRPs related to transporting persons between locations are referred to as Dial-a-Ride Problems (DARP).

According to Psaraftis (1988), VRP-related applications often include two additional important dimensions: a) e*volution* of information, which relates to the fact that the information available to planners might change during the execution of planned routes (e.g. arrival of new customers), and b) type (*quality*) of information, i.e., possible uncertainty on the available data (e.g. the demand of a customer is only an estimate of the actual demand).

Information evolution distinguishes a *static* VRP from a *dynamic* one. In static VRPs all input information is known *a priori* and initial vehicle routes do not change during execution. In dynamic VRPs, part of the input is unknown and is gradually revealed to the planners during execution. Information quality distinguishes a *deterministic* VRP from a *stochastic* one. In the latter, some information is probabilistic, especially the information related to customers, travel or service times, and customer demand. Based on these dimensions, VRPs are often classified in the four categories of Table 2.1.

Table 2.1. VRP taxonomy w.r.t. information evolution and quality

		Type (Quality) of information	
		Deterministic	Stochastic
Evolution of information	Static (Input known *a priori*)	Static & Deterministic	Static & Stochastic
	Dynamic (Input changes over time)	Dynamic & Deterministic (also, *online* or *real-time*)	Dynamic & Stochastic

The problem setting and approach of this dissertation relate to deterministic information only, thus we review below static and dynamic VRPs with deterministic input. For stochastic VRPs, the reader may refer to: a) Bertsimas and Simchi-Levi (1996), Cordeau *et al.* (2007) and Gendreau *et al.* (1996) for the *static* case, and b) Powel (1996), Bent and Van Hentenryck (2004), Larsen *et al.* (2004) and Ichoua *et al.* (2006) for the *dynamic* case.

2.2 Related Static Vehicle Routing Problems

The review of this Section focuses on static VRPs related to the current work. Section 2.2.1 overviews VRPTW, which is the most common static VRP variant; Section 2.2.2 presents the VRPB, the generalized static version of the DVRPMB studied in this dissertation. Finally, Section 2.2.3 drills-down to PDPs, which are also highly related to the static version of the DVRPMB.

2.2.1 The Vehicle Routing Problem with Time Windows (VRPTW)

VRPTW can be defined as follows: A fleet of homogeneous vehicles located at a central depot is tasked to serve a set of customers, each with known demand. A customer can only be served once and within a pre-specified time interval. In the hard TW variant, the customer must be visited after the opening time of this time window, and before its closing time; a vehicle may wait if it arrives to the customer prior to the opening time. In the soft TW variant, the customer may be served outside its time window, but a penalty is added to the objective function. Vehicle capacity cannot be exceeded. The objective of VRPTW is to minimize the total working time (i.e. the sum of travel and waiting times).

The work of Solomon (1987) is one of earliest attempts to tackle VRPTW. The author proposed appropriate conditions for evaluating TW feasibility when a new customer is inserted in a route in order to increase computational efficiency. He also proposed and compared several heuristics for VRPTW. The author found that one of the sequential insertion heuristics (known as Solomon's I1 insertion heuristic), performed particularly well. This heuristic remains a benchmark for VRPTW solution methods.

Over the last decades numerous solution methods have been proposed for addressing VRPTW, ranging from exact approaches to heuristic and metaheuristic methods. Table 2.2 summarizes the most important solution approaches, along with selected references from the literature.

Table 2.2. Types of solution methods for the VRPTW

Solution Method	Algorithm	Reference
Exact	Dynamic programming	Kolen *et al.* (1987)
	Langrangian relaxation	Fisher (1994) Fisher *et al.* (1997) Kohl and Madsen (1997) Kallehauge *et al.* (2006)
	Column generation	Desrochers *et al.* (1992) Kohl *et al.* (1999) Danna and Le Pape (2003) Feillet *et al.* (2005) Chabrier (2006)
	Branch-and-cut	Bard *et al.* (2002)
Heuristic	Construction	Solomon (1986, 1987) Potvin and Rousseau (1993) Ioannou *et al.* (2001) Braysy and Gendreau (2005a)
	Route-improvement	Russell (1977) Baker and Schaffer (1986) Solomon *et al.* (1988) Savelsbergh (1985, 1990, 1992) Potvin and Rousseau (1995)
	Construction & improvement	Russell (1995) Cordone and Calvo (1997) Braysy (2002)
Metaheuristic	Simulated annealing	Chiang and Russell (1996) Tan *et al.* (2001)
	Tabu search	Garcia *et al.* (1994) Potvin *et al.* (1996) Taillard *et al.* (1997) Badeau *et al.* (1997) Chiang and Russell (1997) Cordeau *et al.* (2004) Pisinger and Ropke (2007) Braysy and Gendreau (2005b)
	Evolution (genetic) algorithms	Berger *et al.* (2003) Homberger and Gehring (2005) Mester and Braysy (2005) Mester *et al.* (2007)

Solution Method	Algorithm	Reference
	Ant colony optimization	Gambardella *et al.* (1999)
	Greedy Randomized Adaptive Search Procedure (GRASP)	Kontoravdis and Bard (1995) Chaovalitwongse *et al.* (2003)
	Variable Neighborhood Search	Rousseau *et al.* (2002) Braysy (2003) Braysy *et al.* (2004)

2.2.2 The Vehicle Routing Problem with Backhauls

In the VRP with Backhauls (VRPB), the demand of each customer corresponds to either a delivery (linehaul) or pick-up (backhaul), in which the related items need to be brought back to depot. Typically, VRPB is extended to consider time-windows (VRPBTW). The goal of VRPB is to minimize total travel distance in order to satisfy all delivery and collection requirements. This is typically combined with minimizing the total number of vehicles used. The VRPB may be also viewed as a special case of the Pickup and Delivery Problem (see Section 2.2.3).

There are two main *backhauling strategies* found in the literature that fit the aforementioned problem setting; the *Vehicle Routing Problem with Clustered Backhauls and Time Windows* (VRPCBTW; Gelinas *et al.*, 1995), and the *Vehicle Routing Problem with Mixed Backhauls and Time Windows* (VRPMBTW; Kontoravdis and Bard, 1995). The former (VRPCBTW) imposes visiting sequence restrictions, i.e. all linehaul customers of a route must be served prior to backhaul customers. From a practical perspective, this restriction is used to eliminate rearrangements of load within the vehicle, since normally vehicles are loaded according to the delivery sequence they follow. In case sequencing imposes no priorities, then linehaul and backhaul customers may be visited arbitrarily, as in VRPMBTW, which corresponds to the static version of the problem addressed in the current dissertation.

Several heuristic and exact algorithms have been proposed to tackle the aforementioned problems. Yano *et al.* (1987) proposed one of the first exact approaches to address the VRPCBTW. The authors addressed a case of retail stores, in which the number of pick-up and delivery customers in a route is limited (≤ 4), and developed a Branch and Bound (B&B) algorithm to solve it. Derigs and Metz (1992) investigated a VRPCBTW problem arising in express mail services with up to 80 customers, and proposed various mathematical formulations. Gelinas *et al.* (1995) developed a B&B algorithm based on column generation using a set partitioning model for the VRPCBTW. The authors employed branching on resource variables (time and capacity) instead of network flow variables, which allowed them to solve to optimality a series of Solomon (1987) based problems of up to 100 customers. Toth and Vigo (1997) presented another B&B scheme to solve the symmetric and asymmetric VRPCB, using a Langrangian lower bound by adding cuts, combined with a lower bound that results from relaxing the capacity constraints. Mingozzi *et al.* (1999) also proposed an efficient set-partitioning based integer linear programming formulation for the VRPCB, capable to solve to optimality instances with up to 100 customers and 12 vehicles.

Several heuristics and metaheuristics have also been put forth for tackling VRPB. Table 2.3 summarizes references related to heuristic and metaheuristic approaches proposed to address the VRPCB and VRPMB. Complementary to our review, the reader is also referred to the survey of Parragh *et al.* (2008) for related work on the formulation of these problems, and to the work of Tarantilis *et al.* (2013) for computational results and comparison of various solution approaches on benchmark data sets.

Table 2.3. Types of solution methods for the VRPB

Solution Method	Algorithm	Reference
VRPCB(TW)	Heuristic	Deif and Bodin (1984) Goetschalckx and Jacobs-Blecha (1989) Thangiah *et al.* (1996) Potvin *et al.* (1996) Toth and Vigo (1999)
	Metaheuristic	Duhamel *et al.* (1997) Hasama *et al.* (1998) Reimann *et al.* (2002) Osman and Wassan (2002) Zhong and Cole (2005) Brandao (2006) Reimann and Ulrich (2006) Ropke and Pisinger (2006) Zachariadis *et al.* (2012) Tarantilis *et al.* (2013)
VRPMB(TW)	Heuristic	Golden *et al.* (1988) Casco *et al.* (1988) Kontoravdis and Bard (1995) Salhi and Nagy (1999) Wade and Salhi (2002)
	Metaheuristic	Hasama *et al.* (1998) Zhong and Cole (2005) Reimann and Ulrich (2006) Ropke and Pisinger (2006) Tarantilis *et al.* (2013)

2.2.3 The Pickup and Delivery Problem

Pickup and delivery problems (PDPs) form a class of vehicle routing problems in which goods or passengers are transported between an origin and a destination. Therefore, each transportation request i is associated with two vertices, p_i and d_i, and the goods (or passengers) should be picked up at p_i and delivered to d_i. For a solution to be feasible in this setting, p_i and d_i should be included in the same route, and p_i should be visited prior to d_i. Typically, capacity constraints are considered, and a time window is associated with each vertex. A characteristic example of time window and capacity constraints can be found in applications related to “Transportation on Demand” (Cordeau *et al.*, 2007), which involve the transportation of people with special needs (Dial-a-Ride Problem, DARP).

Berbeglia *et al.* (2008, 2010) introduced a classification scheme for PDPs based on the number of origins and destinations involved. Based on this scheme, PDPs can be classified into three (3) different categories:

a) *Many-to-Many (M-M)* problems, in which any vertex may serve as source or a destination,
b) *One-to-Many-to-One (1-M-1)* problems, in which goods/passengers initially available at a depot are to be transported to multiple sites, while other goods/passengers available at these or other sites need to be transported back to the depot, and
c) *One-to-One (1-1)* problems, in which each item/passenger is associated with a certain origin and a certain destination.

These problem types are reviewed below.

2.2.3.1 Many-to-Many (M-M) PDPs

According to Berbeglia *et al.* (2008), a typical example of the first category (M-M problems) is the so-called *Swapping Problem* (Anily and Hassin, 1992). In this problem, each vertex initially possesses an object of a known type, and requests an object of a desired type. The objective is to construct the pick-up and delivery routes in such a way that every vertex will eventually possess an object of the desired type. Interested readers may refer to Anily and Hassin (1992), Anily *et al.* (2006) and Wang *et al.* (2006).

2.2.3.2 One-to-Many-to-One (1-M-1) PDPs

In this class of problems, which constitutes a generalization of VRPB, some customers require delivery of commodities located at a depot (referred to as linehauls), while other customers require pick-up of commodities from their sites and delivery to the depot (referred to as backhauls). In terms of demand, there are two variants of 1-M-1 PDPs: *combined* demand and *single* demand. The former relates to cases in which a customer requests both pick-up and delivery operations (simultaneously), while the latter relates to cases in which each customer requests either pick-up or delivery, as in the VRPB.

A solution is said to be *mixed* if pick-up and delivery customers may be served in an arbitrary order; then, the related problem is called the *1-M-1 PDP with Single Demands and Mixed Solutions* and is equivalent to the VRPMB described in Section 2.2.2. On the other hand, the VRPCB is equivalent to the *1-M-1 PDP with Single Demands and Backhauls*.

As explained earlier, the 1-M-1 PDP with single demands forms a generalization of the VRPB described in Section 2.2.2; thus, the references surveyed in that Section also apply in this case. For the combined demand case (which is not relevant to this dissertation), we refer interested readers to the review of Berbeglia *et al.* (2008) and the work of Gribkovskaia *et al.* (2007), Gribkovskaia and Laporte (2008), Chen and Wu (2006) and Bianchessi and Righini (2007).

2.2.3.3 One-to-One (1-1) PDPs

As already mentioned, 1-1 PDPs relate to problems in which each commodity is associated with exactly one pick-up and one delivery vertex (location). This type of problems can also be referred to as paired-PDPs. Two main variations can be found: a) the *VRP with Pickups and Deliveries* (VRPPD), which is related to the transportation of goods, and b) the *Dial-a-Ride Problem* (DARP), which deals with the transportation of people. The main difference of these two variants is that in the latter, passenger convenience is usually taken into account. Recently, researchers focused in another category of this type of problems: the *VRPPD with Transshipments* (VRPPDT), in which vehicles are allowed to drop off goods/passengers to intermediate locations to be picked up and delivered to the final destination by another vehicle. This class of problems is relevant to the problem addressed in Chapter 7 of this dissertation, and the related literature is discussed in Section 7.2.

2.3 Dynamic Vehicle Routing Problems

The Dynamic Vehicle Routing Problem (DVRP) is the dynamic counterpart of the classical 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 has focused more and more on dynamic problems, and various related models and algorithms have been developed. Rapid growth in telecommunications and information technology have led to this direction, since, distribution companies are now able to monitor the vehicles' location and status in real-time, and, thus, to manage them in real time. Related applications of DVRP include courier service systems, dial-a-ride systems, emergency systems, etc.

Figure 2.3 presents a simple example of a dynamic vehicle routing situation. In this example, two vehicles must serve static orders that are known *a priori* (represented by closed black nodes), as well as dynamic orders (represented by open nodes) that are revealed during the execution of the initial routing plan. The latter is presented in Fig. 2.3a; Figure 2.3b presents an intermediate state in which the vehicles have already executed a part of the plan (denoted by a dashed line) and are on their way to their next destinations (denoted by thick continuous line). Between the departure of the vehicles from the depot and the time related to Fig. 2.3b, two dynamic orders (DO) have arrived and need to be incorporated in the current plan. In Fig. 2.3c, a DO has been successfully incorporated in the plan without significant cost or delay (route on top of the Figure). However, incorporating the second DO in the bottom route will cause a large detour, illustrating the complexity of incorporating new requests in the plan.

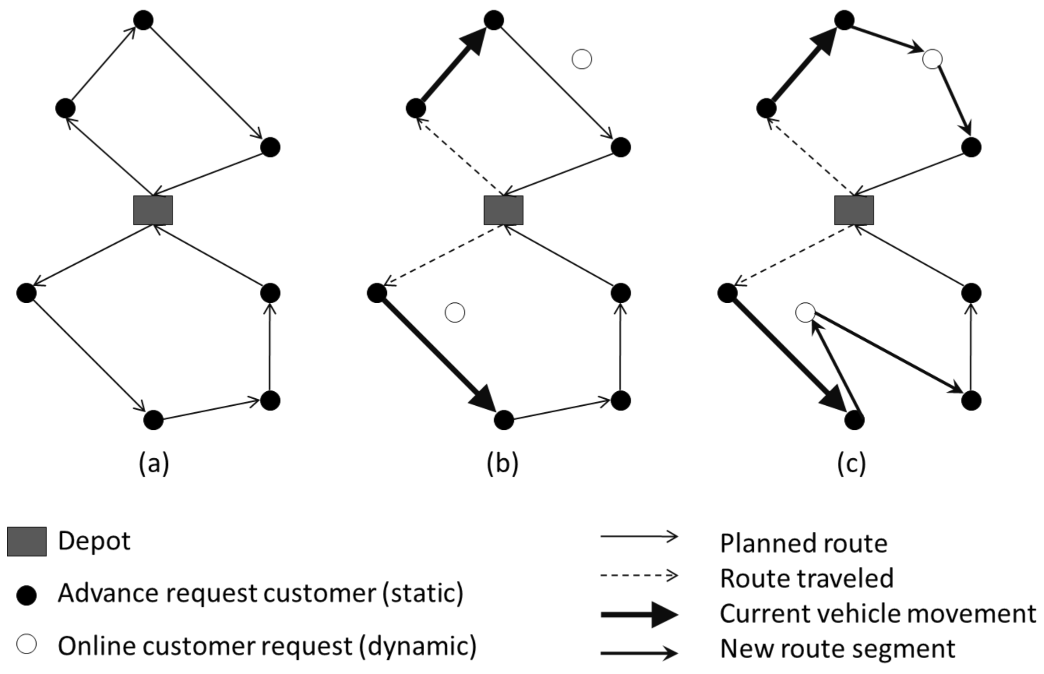


Figure 2.3: A dynamic vehicle routing scenario: (a) initial routing solution, (b) state after DO arrival, (c) incorporation of DO into the plan

The most common source of dynamism in VRPs is the arrival of customer requests during the execution of the routing plan. Another important dynamic component of practical applications is related to variable travel times (e.g. time-dependent) and variable onsite service times

(Fleischmann *et al.*, 2004; Chen *et al.*, 2006; Potvin *et al.*, 2006; Guner *et al.*, 2012, Haghani and Jung, 2005). Recent work has also considered the case in which customer demand is revealed when the vehicle reaches the customer site (Tatarakis and Minis, 2009; Novoa, 2009; Secomandri, 2009), as well as the case in which a vehicle executing a route becomes unavailable (e.g. vehicle breakdowns, Li *et al.*, 2009a; 2009b, Mamassis *et al.*, 2013; Mu *et al.* 2011).

The source of dynamism for the problem addressed in this dissertation is the arrival of dynamic requests; therefore, in the following, we focus our review on this aspect.

According to Larsen *et al.* (2007) the DVRP has two main differences w.r.t. the static VRP: a) not all information relevant to planning the routes is known by the planner when the routing process begins, b) information may change after the initial routes have been designed. DVRP is a more elaborate and complex problem than its static counterpart, and belongs to the class of NP-hard optimization problems (Psaraftis, 1988). As a result, it is not always feasible to obtain optimal solutions to problems of practical size within a reasonable timeframe.

As mentioned above, DVRPs introduce new elements and challenges that increase the complexity of the related routing decisions. For example, in some contexts, such as in express (same-day) courier services, the company may deny a service request either because of significant high service costs or because it is impossible to serve the particular request (Ichoua *et al.*, 2000; 2003; 2006). DVRPs also employ different objective functions. A common objective in static VRPs is the minimization of routing cost, while DVRPs may introduce additional elements such as service level, service or profit maximization, response times, etc. Finally, DVRPs require online decision making, which may compromise reactiveness (to input changes) in the light of decision quality (e.g. lower costs). The best trade-off between reactiveness and decision quality can also be an aspect of particular importance in many applications in which customers call for service (e.g. repair-maintenance services).

2.3.1 Measuring dynamism (degree of dynamism)

According to Ichoua *et al.* (2007), the dynamism of a problem may be characterized by two elements: a) the *frequency of changes*, i.e. the rate at which new information becomes available, and b) the *urgency of requests*, which is the elapsed time between the arrival of a new request and its required service time. Based on these aspects, three metrics have been proposed to measure the dynamism of a problem:

(i) Lund *et al.* (1996) defined the *degree of dynamism* δ as the ratio between the number of dynamic requests n_d and the total number of requests n_{tot}:

$$\delta = \frac{n_d}{n_{tot}} \quad (2.1)$$

(ii) Larsen (2001) generalized δ in order to take into account the arrival time of requests and proposed the *effective degree of dynamism*, δ^e. Let T_{max} be the length of the planning horizon, N the set of requests (n_{tot} in total), and h_i the disclosure time of request i (operations start at time 0); then δ^e can be defined as:

$$\delta^e = \frac{1}{n_{tot}} \sum_{i \in N} \frac{h_i}{T_{max}} \quad (2.2)$$

(iii) Finally, Larsen (2001) extended δ^e to problems with time windows in order to consider also the *urgency* of requests. The author defines the *reaction time*, as the difference between the closing of the time window b_i of request i and the disclosure time h_i; longer reaction times denote higher flexibility to include a request in the current routes. Thus, the effective degree of dynamism is extended as follows:

$$\delta^e_{TW} = \frac{1}{n_{tot}} \sum_{i \in N} \left(1 - \frac{b_i - h_i}{T_{max}}\right) \quad (2.3)$$

It should be noted that the aforementioned metrics assume values in the interval $[0,1]$, and higher values within this interval denote higher level of dynamism.

2.3.2 Classification of DVRPs

DVRPs typically follow the taxonomy of VRPs, i.e. every VRP or PDP variant discussed in Section 2.2 may be related to a dynamic counterpart, in case portion of the related data is not known in advance and is revealed over time.

In addition to classifications that follow the taxonomy of static VRP problems, DVRPs are typically classified according to *dynamism*, i.e. the extent of dynamic information with respect to static information (the information known prior to the start of operations). Larsen *et al.* (2002; 2008) used the effective degree of dynamism (δ^e) to support such a classification. Prior to discussing their proposal, we discuss typical objectives employed in DVRPs considering the degree of dynamism. These objectives include the following:

1) **Transportation costs**: This objective used extensively in static routing should also be considered in DVRP systems, due to the importance of transportation.
2) **Service maximization**: Maximizing the number of dynamic orders served is relevant and significant in those DVRP systems which are not capable of serving all dynamic orders (e.g. in case of limited resources).

3) **Responsiveness**: Offering higher service level to customers may not be compatible with cost minimization, since a prompt response to a new dynamic request may imply sub-optimal routing of the vehicle in terms of distance/cost.

Using δ^e, Larsen *et al.* (2002; 2008) classified DVRPs to weakly, moderately and strongly dynamic systems:

- **Weak dynamic systems ($0 < \delta^e \leq 0.3$)**: In those systems, only a limited number of customer orders is revealed dynamically, while most orders are known prior to the start of execution. The typical objective employed in this case is to minimize transportation costs.
- **Moderate dynamic systems ($0.3 < \delta^e \leq 0.8$)**: In this case, the proportion of dynamic orders is significant, but static orders should also be considered during the design of the initial plan. The typical objective here is a combination of cost minimization and maximization of dynamic orders served.
- **Highly dynamic systems ($0.8 < \delta^e \leq 1$)**: It is the most extreme case of dynamic routing systems, met mainly in emergency services such as police, fire department and ambulance services. Here, almost no requests are known in advance and the routing plan is constantly changing (in a real-time fashion) based on the newly received requests. These applications are characterized by a strong focus on responsiveness (or service maximization).

The DVRP variants (problems) investigated in this dissertation are related to moderate dynamic systems. In particular, we investigate cases in which the total number of dynamic orders comprises a significant portion of the total number of orders to be served. Consequently, we focus on minimizing the total transportation costs when in case there are sufficient vehicles available to serve all orders (Chapters 5 and 7). For the case of limited fleet, we consider maximizing the number of dynamic orders served (Chapter 6).

2.3.3 A review of DVRP applications

DVRP research has also been inspired by applications (Pillac *et al.*, 2013) in: i) on-site service delivery (e.g. maintenance), ii) transport of goods, and iii) transport of persons.

In the category of onsite service delivery, a request is defined by a customer location and, oftentimes, a time-window. A typical application concerns the area of maintenance operations, in which companies offer scheduled periodic maintenance visits (planned offline), as well as corrective maintenance on short notice (planned online). Therefore, each technician starts a certain route at the beginning of the day, while new requests have to be incorporated dynamically throughout the day. Vehicle capacity is oftentimes not an issue in this case. Studies related to onsite service delivery can be found in Larsen *et al.*, 2004, Bertsimas and Van Ryzin, 1991, Beaudry *et al.*, 2010 and Gendreau *et al.*, 1999.

Related research on transportation of goods has normally addressed transportation within urban areas, typically referred to as *city logistics*. These cases are characterized by highly unpredictable travel times (Zeimpekis *et al.*, 2007) and other aspects, including collaboration between companies to take advantage of economies of scale. One common application concerns courier services. In this setting, couriers are dispatched to deliver packages to customers while new requests are received in real-time through a call-center. Those new arriving requests have to be collected from the customer location and either delivered to a desired destination or to a unique depot. The problem then is to dynamically route vehicles taking into account the requests known prior to the start of operations, as well as the newly received (dynamic) ones; other dynamic information might be relevant in this environment, such as traffic conditions and variable travel times. Related studies include those by Gendreau *et al.* (2006), Ghiani *et al.* (2009), Attanasio *et al.* (2007) and Angelelli *et al.* (2009). Other practical settings with similar characteristics include the delivery of press media (Bieding *et al.*, 2009; Ferucci *et al.*, 2013), grocery delivery services (Campbell and Savelsbergh, 2005; Azi *et al.*, 2014), and transport of goods in warehouses and hospitals (Fiegl and Pontow, 2009).

Finally, applications related to transportation of passengers bear similarities to the transport of goods, although they include additional constraints related to service levels, such as passenger inconvenience (waiting, travel, and service times). Typical applications comprise planning of taxi services (Caramia *et al.*, 2002; Fabri and Recht, 2006), transportation of children, patients, elderly or disabled people (Cordeau *et al.*, 2007; Berbeglia *et al.*, 2010).

The research in this dissertation has been motivated by applications related to *transport of goods* (city logistics), such as courier services, money-transfer operations and repair-maintenance services. The dynamism comes from a single source, namely the occurrence of new service requests (dynamic orders); there is no uncertainty associated with service locations, travel times or traffic conditions.

2.3.4 Significant solution methods

Below we review significant methods and solution approaches for deterministic DVRPs, in which dynamism stems from the arrival of new dynamic requests. The interested reader may also refer to the review work of Gendreau and Potvin (1998, 2004), Ghiani *et al.* (2003), Zeimpekis *et al.* (2007), Ichoua *et al.* (2007), Goel (2008), Larsen *et al.* (2008) and Pillac *et al.* (2013). For the case of stochastic DVRPs, we refer the reader to Powel (1996), Bent and Van Hentenryck (2004), Larsen *et al.* (2004), and Ichoua *et al.* (2006).

In dynamic deterministic routing problems, the information is gradually revealed over time. Consequently, exact approaches may provide an optimal solution only for the current state and cannot guarantee that the overall solution will remain optimal, or even efficient, once new input is introduced. Therefore, most approaches employ heuristics that can quickly provide a solution which incorporates the up-to-date information.

According to Pillac *et al.* (2013), the approaches for addressing deterministic DVRPs can be classified in two main categories: a) approaches that apply *periodic re-optimization*, and b) approaches which are based on *continuous re-optimization*. Those are described in the following paragraphs. In addition, we overview some *advanced* approaches.

2.3.4.1 Periodic re-optimization approaches

In this, the most common, solution strategy an efficient VRP algorithm is used or adapted to solve the static version of the problem at selected multiple times. Periodic re-optimization approaches typically commence at the beginning of the day with an initial optimization that produces an initial set of routes, either for orders known prior to the start of execution (for weak to moderate dynamic systems) or for the first (dynamic) orders received (for highly dynamic systems). Then the solution is re-optimized either whenever the available information changes, or at fixed re-optimization intervals, normally referred to as decision epochs (Chen and Xu, 2006), or time slices (Kilby *et al.*, 1998). A wide variety of algorithms may be used for re-optimization, ranging from simple policy-based techniques and heuristics to exact algorithms; however, computational effort is of significant importance here.

Psaraftis (1980) was the first to apply periodic re-optimization for a dynamic Dial-a-Ride problem. The author proposed a dynamic programming approach in order to find the optimal route each time a new request was received, and was able to solve problems with relatively small number of requests.

Yang *et al.* (2004) proposed a linear programming approach that is applied whenever a new request is received for the real-time truckload PDP. Chen and Xu (2006) proposed a column-generation-based approach for solving a DVRP with hard time windows, in which all requests need to be serviced; the algorithm uses fast heuristics to modify existing columns generated at an earlier stage in order to incorporate the up-to-date information. Those columns are then included in the Restricted Master Problem (set-partitioning formulation), which is solved, and the process is repeated in an iterative manner. Their approach outperformed in solution quality an insertion-based heuristic used for comparison, but provided inferior results compared to a similar approach that allows unlimited computational time for solving the underlying static problems.

Shieh and May (1998) studied the DVRP with time windows and proposed for each re-optimization step an insertion-based heuristic followed by a local search. Larsen *et al.* (2002) compared various rule-based heuristics on instances with various *degrees of dynamism* for a dynamic travelling repairman problem, in which requests need to be served at minimum total cost. Their study illustrated that the route length increases linearly w.r.t. the degree of dynamism. Montenammi *et al.* (2005) employed an Ant Colony System (ACS) in order to solve the dynamic VRP by dividing the overall planning horizon in periods (time-slices), as in Kilby *et al.* (1998). During each time-slice a static optimization problem is solved by considering all requests known at the beginning of this time slice. A similar approach was also employed by Gambardella *et al.* (2003) and Rizzoli *et al.* (2007).

It should be emphasized that the quality of the solution of the overall problem through multiple re-optimization steps is highly dependent on the solution approach employed at each step. On one hand, inferior re-optimization results at each step (e.g. obtained through simple heuristics) may lead to significantly inferior solutions. On the other hand, even the use of an exact algorithm cannot guarantee the generation of superior solutions for the entire problem. The computational results of Yang *et al.* (2004) and Chen and Xu (2006) concerning a range of dynamic routing problems, indicated that employing mathematical-programming-based approaches over simple ones may indeed yield better overall solutions.

In addition to the problem definition and the solution approach, a critical problem element is when to *re-optimize*. Very limited research has focused on re-optimization policies and their impact on the overall solution. The majority of studies (e.g. Gendreau *et al.*, 1999; Ichoua *et al.*, 2000) *re-optimize* at every *event*, i.e. upon the arrival of a vehicle to a customer, or the introduction (or cancellation) of a customer order. Other studies deal with re-optimization at certain fixed periods. For example, Larsen (2001) studied the DVRP with time windows introducing the so-called *batching strategies*, and analyzed the effect of re-optimization on simple predefined fixed events (e.g. upon the arrival of three customers and every 10 minutes). Chen and Xu (2006) *re-optimize* at fixed cycles. More recently, Angelelli (2009) applied re-optimization on predefined time-events (e.g. 1, 2.5 and 5 hours) for a dynamic multi-period vehicle problem.

2.3.4.2 Continuous re-optimization approaches

In this type of approach, re-optimization is being performed continuously throughout the course of operations, based on the current state of the system and past knowledge.

In particular, the approach commences at the beginning of operations with an initial set of routes (as in periodic re-optimization), and vehicles are informed only about their next destination. An elaborate mechanism is continuously executed (in the background) to further improve the solution according to the currently known state of the system, and stores good quality solutions consistent with this state in a memory that is adapted continuously (*adaptive memory*, Taillard *et al.*, 2001). A decision procedure is used to update the solutions in the adaptive memory whenever the available information is updated. Updates typically occur due to a) service completion at a customer location, or b) the occurrence of a new (dynamic) request. For the updates that relate to service completion, the decision procedure identifies the next destination of the vehicle based on the best solution stored in the adaptive memory. When a new request occurs, the request is inserted (e.g. using a local search heuristic) in each solution stored in the adaptive memory, and, thus, all existing solutions are updated. As long as there are no incoming requests and no services are completed, the mechanism keeps running in an attempt to improve the routes in the adaptive memory. The latter is an important advantage for this approach.

Gendreau *et al.* (1999) were the first to employ continuous re-optimization. The authors proposed a tabu search heuristic similar to the one introduced by Taillard *et al.* (1997) in order to address the DVRPTW arising in a long-distance courier service, in which time windows may be violated at some cost. No stochastic (forecasting) information about incoming (dynamic) requests has been assumed. Their approach maintains a pool of good solutions (routes) based on the available data in an adaptive memory which is used to generate initial solutions for a parallel tabu search. When a new request is received, it is inserted into each solution residing in the adaptive memory through a cheapest insertion process in order to decide whether to accept or reject the request. The best solution is selected after applying a fast local search procedure. The solutions in the adaptive memory are also updated upon service completion at a customer location. A similar approach has been employed by Ichoua *et al.* (2000, 2003) for the DVRP, by Gendreau *et al.* (2006) and Chang *et al.* (2003) for the DPDP, and Attanasio *et al.* (2004) for DARP.

Another tabu search approach with adaptive memory has been employed by Bent and Van Hentenryck (2004), who introduced the concept of the Multiple Plan Approach (MPA). The latter attempts to continuously generate different solutions, which incorporate both static and known dynamic requests. Under this concept, a pool of solutions (routing *plans*) is used to generate a so-called *distinguished plan*. Upon arrival of a new request, a mechanism (e.g. a local search heuristic) checks whether the request can be incorporated or not in the current pool of solutions; if yes, the request is incorporated in the solution pool (routing plans) and solutions from the pool that cannot fit this request are discarded. The pool of solutions is updated during each event in order to ensure that all solutions are consistent with the current state of the system.

Finally, Genetic Algorithms (GA) have been also used in continuous re-optimization. GA algorithms in dynamic contexts are very similar to those designed for static problems, although they generally run throughout the planning horizon and solutions are constantly adapting to the input changes. The interested reader can refer to the work of Benyahia and Potvin (1998), Cheung *et al.* (2008), and Van Hemert and Poutre (2004).

2.3.4.3 Advanced strategies

The last type of approaches includes advanced methods that exploit the nature of dynamic problems. For example, Ichoua *et al.* (2000), motivated from courier applications, proposed a new method for the dynamic assignment of new requests, in which a vehicle may be diverted from its next destination in order to serve a new request. The method is integrated in the tabu search framework of Gendreau *et al.* (1999), and the authors demonstrate through computational experiments that this strategy yields a reduction in the total distance traveled, compared to the case in which the vehicle may not be diverted from its next destination.

Other studies have introduced waiting strategies, which consider the possibility of positioning vehicles at strategic locations, or at customer sites, in order to wait for the arrival of potential new (dynamic) requests (see Branke *et al.*, 2005; Mitrovic-Minic and Laporte, 2004; Ichoua *et al.*, 2006).

2.3.5 Performance assessment

Measuring the performance of the solution of a dynamic optimization problem, such as the one addressed here, is not a straightforward task. The literature has suggested that new metrics are required for this task (Mitrovic-Minic *et al.*, 2004; Pillac *et al.*, 2013).

Sleator and Tarjan (1985) introduced *competitive analysis* (see also Borodin and El-Yaniv; 2005, Jaillet and Wagner; 2008, Larsen *et al.*; 2007). Consider a problem instance I in which data is revealed in real-time, and its offline counterpart I_{off} in which all data (of instance I) are available beforehand (prior to constructing the solution). Let $z^*(I_{\text{off}})$ be the cost of the optimal solution of I_{off}. Also, consider an algorithm A solving I. Let $z_A(I)$ be the cost of the solution obtained by A for instance I. Then, algorithm A is said to be $c-competitive$ if there is a constant α such that:

$$z_A(I) \leq c \cdot z^*(I_{\text{off}}) + a \qquad \forall I \in \ddot{I} \tag{2.4}$$

If a is equal to zero, then the algorithm is said to be *strictly-c-competitive*, meaning that the value of the objective function of the solution determined by A for instance I will be at most of c times greater than the optimal value. For example, a *strictly-2-competitive* dynamic algorithm guarantees that the value of the solution would never be more than twice the value of the optimal solution of the static problem (for any investigated instance). Thus, competitive analysis offers a worst-case measure of performance.

However, the competitive analysis metric assumes that Ineq. (2.4) should be explicitly proven, which in many cases (except very simple ones) is not possible.

The *value of information* originally introduced by Mitrovic-Minic *et al.* (2004), provides a more practical metric. Consider a DVRP instance $\mathcal{H}$ and the related static problem $\mathcal{H}_s$, in which all dynamic information is known prior to dispatching the vehicles (i.e. at time $t = 0$). Then the value of information metric $V_{\mathcal{F}}$ corresponding to algorithm $\mathcal{F}$ while solving dynamic problem $\mathcal{H}$ is defined by the following expression

$$V_{\mathcal{F}}(\mathcal{H}) = \frac{z_{\mathcal{F}}(\mathcal{H}) - z_{\mathcal{F}}(\mathcal{H}_s)}{z_{\mathcal{F}}(\mathcal{H}_s)} \times 100 \tag{2.5}$$

where $z_{\mathcal{F}}(\mathcal{H})$ and $z_{\mathcal{F}}(\mathcal{H}_s)$ are the values of the objective function for dynamic problem $\mathcal{H}$ and for the related static problem $\mathcal{H}_s$, both solved by algorithm $\mathcal{F}$. Note that $\mathcal{F}$ is employed at each re-optimization step for $\mathcal{H}$, while $\mathcal{F}$ is employed once to solve $\mathcal{H}_s$.

2.3.6 Dynamic Pickup and Delivery Problems (DPDP)

Limited work has been conducted on the dynamic counterpart of the PDPs. DPDPs can be classified along the lines discussed in Section 2.3.3. To the best of our knowledge, only the work of Chang *et al.* (2003) and Wang and Cao (2008) have investigated the dynamic version of 1-M-1-PDPs. In particular, Wang and Cao (2008) addressed a Dynamic VRPCBTW with demand changes. The authors identify which demand changes disrupt the original plan and propose a disruption recovery model based on a local-search algorithm. They used a small scale example to illustrate their model's ability to achieve savings in disruption situations. Chang *et al.* (2003) addressed the real-time VRPTW with simultaneous pick-up/delivery demands (RT-VRPTWDP) and formulated it as a mixed-integer programming model. They proposed a tabu search algorithm to solve the problem every time a new request is received or altered. Their method outperformed simple route construction and improvement approaches on the 15 benchmark instances of Gelinas *et al.* (1995).

The majority of existing work has focused on dynamic one-to-one PDPs (1-1 DPDPs), in which each request has certain origin and destination. 1-1 DPDPs mostly deal with the transportation of passengers in urban areas, as in the *dial-a-ride problem* (DARP), or in the same-day transportation of letters/parcels, referred to as *Dynamic PDP* (DPDP). For this class of problems we refer the reader to the survey of Berbeglia *et al.* (2010), which overviews solution approaches and related studies.

2.4 Branch and Price through Column Generation

The main technique employed in this dissertation to tackle DVRPMB is based on the Branch-and-Price method, which is reviewed briefly below.

Branch-and-Price consists of a column generation algorithm embedded within a branch-and-bound scheme (Barnhart *et al.*, 1998; Desaulniers *et al.*, 1998; Desrosiers and Lübbecke, 2005). 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 (CG) is regarded as one of the most promising methods to solve vehicle routing problems by finding "good" lower bounds, especially when the objective is to minimize the cost (normally the distance travelled). 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 (columns) 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 repeatedly using a limited set of possible "good" routes, which are generated by solving a series of simpler sub-problems.

Specifically, by solving the set partitioning 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 may further reduce the value of the objective function. This is the column generation step. The values of the optimal dual variables provided by the restricted problem are incorporated as modified costs in the objective function of the sub-problems (usually simpler optimization problems), which, in turn, provide promising new routes (i.e. routes with negative reduced costs) that should be included in the formulation. Subsequently, the linear relaxation of this expanded problem is resolved. This process is performed iteratively until no other columns may be found to reduce the value of the objective function.

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 $\mathcal{H}$ (in our case $\mathcal{R}'$ is a subset of all possible feasible routes)

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

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

Step 4. For $r \in \mathcal{R}$ with $\bar{c}_r < 0$ add 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.

It is clear that the speed of convergence of the CG algorithm depends mostly on the column generation step (Step 3). If the optimal solution is pursued, then an exact algorithm may be used for this step (e.g. solution of a Shortest Path Problem with dynamic programming); otherwise, powerful heuristics and/or metaheuristics could be used in order to provide a sufficient trade-off between solution quality and computational time. Depending on the algorithm used, a large number of columns with negative reduced cost may be generated at each step, in order to converge to a solution in fewer iterations.

2.5 Dissertation objectives and contribution

In this dissertation we focus on the deterministic version of dynamic 1-M-1 PDPs which have yet to be investigated as discussed in Section 2.3.6. In particular, we focus on a variant of dynamic 1-M-1 PDP which is equivalent to the dynamic version of the Vehicle Routing Problem with Mixed Backhauls (DVRPMB). To the best of our knowledge, this problem has not been investigated in the literature.

Although considerable progress has recently been made in studying dynamic vehicle routing problems, key issues remain to be investigated with significant implications to both the theoretical treatment of the underlying problems and the related application of the proposed approaches. Within the context of DVRPMB, we attempt to address some of these key issues (research questions):

- For this type of problems, are there exact, or near-optimal, methods that may solve the static re-optimization problem in a time-efficient manner (suitable for a real time environment)?
- Within the re-optimization framework, what is the appropriate sequence of time instances (re-optimization schedule) to invoke the re-optimization method in order to obtain superior solutions to the entire problem? Which factors of the environment affect the choice of the re-optimization schedule?
- Within the re-optimization framework, what is the appropriate process to release the re-optimized plan (to the fleet), and how does this process affect the quality of the solution of the overall problem?
- Are there fundamental differences between problems that consider unlimited fleet resources, and problems with limited fleet resources? If so, how can one address these differences?
- How can one capitalize on load transfer processes, in order to overcome one of the dominant problem constraints raised by the initial assignment of known (static) orders to vehicles? What are the implications on the formulation of the mathematical model, and on the performance of the system?

By addressing the above research questions, this dissertation makes the following contributions:

1. We propose an appropriate periodic re-optimization process to address DVRPMB and a mathematical formulation for the corresponding re-optimization problem (invoked in each re-optimization cycle). In addition to defining the re-optimization model, we drill-down to significant aspects concerning the re-optimization process; i.e. i) *how* to *re-optimize*, ii) *when* to *re-optimize*, and iii) *which* part of the new plan to communicate to the drivers.
2. Regarding "*how to re-optimize*", we propose an exact approach based on Branch-and-Price (B&P). The contribution of our method compared to typical B&P applications in vehicle

routing problems is two-fold. First, we introduce an appropriate structure that exploits the characteristics of the dynamic problem in hand and solves a series of sub-problems to identify columns that can further enhance the value of the objective function. This decomposition allows the algorithm to be amenable to dynamic problems of practical size, without losing optimality. Secondly, we appropriately enhance the dominance criteria used in the sub-problems in order to ensure optimality in a time-efficient manner; this is achieved by discarding a large number of non-promising paths.

3. In order to address challenging cases (e.g. without time-windows), we propose a novel Column Generation-based insertion heuristic that provides near-optimal solutions in an efficient manner.
4. Regarding "*when to re-optimize*" we present and analyze typical re-optimization policies that consider various re-optimization frequencies. In addition, we investigate the effect of two *implementation tactics*: i) immediate release of all dynamic orders for implementation (Full Release - FR) and, ii) release of only those dynamic orders that are scheduled for implementation prior to the next re-optimization cycle (Partial Release -PR). We provide theoretical insights regarding the expected behavior of those tactics and use extensive experimentation to test the proposed methods and analyze the related re-optimization policies. Based on the results obtained we propose re-optimization guidelines under various operational settings.
5. In order to address significant practical aspects, we modify the DVRPMB model to consider the case of limited fleet (in which not all customer orders can be served within the planning horizon). To address this case, we introduce appropriate objective functions that account for vehicle productivity during each re-optimization cycle, and we illustrate that those objectives may offer higher customer service.
6. We apply our proposed methods for the DVRPMB with limited resources to a large practical case of a next-day courier service provider. Through this case study, we illustrate that our approach outperforms the dispatchers' current practices, as well as a sophisticated insertion-based heuristic used for comparison.
7. We also investigate an interesting problem that attempts to overcome or, at least, moderate the intrinsic constraint of preventing delivery (static) orders to be reassigned to vehicles other than the one originally assigned to. To do so, we examine a policy of transferring (delivery) orders between vehicles during execution of the distribution plan. We incorporate load-transfer operations within the DVRPMB framework; we refer to this problem as

DVRPMB with Load Transfers (DVRPMB-LT). We examine two types of exchange locations (fixed, or at the location of any customer not yet served).

8. We model the re-optimization problem related to DVRPMB-LT using an arc-based formulation in order to be able to provide exact solutions, and compare them to the optimal solutions of the re-optimization problem that does not allow transfers. Furthermore, we develop an appropriate heuristic that is able to address (solve) practical cases with an extended solution space. We illustrate through extensive experimentation that load-transfer operations can offer substantial savings for the overall dynamic problem.

Chapter 3: THE DYNAMIC VEHICLE ROUTING PROBLEM WITH MIXED BACKHAULS

The main scope of this Chapter is to define the Dynamic Vehicle Routing Problem with Mixed Backhauls (DVRPMB), and to set the foundation for the solution approach of Chapter 4. Section 3.1 presents basic characteristics and assumptions of DVRPMB. Section 3.2 provides an overview of the solution framework and the re-optimization problem to be solved in each iteration of this framework. It also presents the mathematical formulation of the re-optimization problem and discusses the problem's complexity.

3.1 Problem description

3.1.1 Problem overview

Consider a transportation network in a Euclidean plane. A sufficient number of homogeneous vehicles (set V) with limited capacity $\bar{Q}$ are located at a single depot prior to the start of operations. At time 0, at the beginning of the planning horizon $[0, T_{max}]$, a set of vehicles $K \subset V$ commence the execution of their planned routes to serve a set of offline requests known in advance (typically requiring delivery services), while $K^d = V - K$ is the set of vehicles available at the depot. A vehicle, once dispatched, is required to return to the depot until $t = T_{max}$. Orders known in advance may require service within a certain time-window, and all information regarding those orders is known prior to the execution of the planned routes. We refer to such orders as *Static Orders, SO*.

During the execution of the distribution plan, new customers call-in, requesting (pick-up) services. These arriving requests (hereafter denoted as *Dynamic Orders*, *DO*) have to be collected and returned back to the depot. Only DO that arrive during a pre-defined *admissible period* $[0, T_{max} - \tau]$ must be served, where τ denotes a predefined time interval (e.g. the last hour of the available working period). Orders arriving at time $t \geq T - \tau$ are deferred to the following day. Static orders originally assigned to vehicles in K cannot be re-allocated to other vehicles, while DO may be served by any vehicle $V = K \cup K^d$ as needed. In general, customer orders in the current context have the following characteristics: i) static orders (SO) may be deliveries or pick-ups, ii) all dynamic orders (DO) are related to pick-up operations, and iii) all DO are returned to the depot for further processing.

The problem's scope is to serve all SO and allocate DO to the vehicles of set V as best as possible. This scope may be formalized according to the availability of the fleet; under this framework, there are two cases to be considered:

i. *Unlimited fleet of vehicles*: Serve all static orders and all DO that arrive within the admissible period $[0, T_{max} - \tau]$, so as to minimize the sum of the total distance traveled by the dispatched vehicles. This case is studied in Chapters 4 and 5.
ii. *Limited fleet of vehicles*: Serve all static orders and maximize the number of served DO throughout the available shift. This case is examined in Chapter 6.

For case (i) above, a sufficient number of homogeneous vehicles are located at a single depot at the beginning of the planning horizon in order for the fleet to serve all orders; thus new vehicles may be dispatched to serve some DO that can't be served by vehicles *en route.* For case (ii), the fleet is sufficient to serve static orders, but may not be sufficient to serve all DO.

These objectives are considered under the following operational constraints:

- All SO should be served
- Each order may be served at most once, by a (single) vehicle
- SO cannot be reassigned among vehicles, i.e. the static orders originally assigned to a vehicle, must be served only by this vehicle. Of course, the sequence of servicing SO by a certain vehicle may be changed, if this favors the objective function.
- The service of an order must commence within a pre-specified time-window, i.e. the service of an order cannot commence prior to the opening of this time-window and after its closure.
- All vehicles should return to the depot within $[0, T_{max}]$, i.e. within the allowable working period (e.g. driver's shift).
- The total load of the vehicle at any time cannot exceed the vehicle's capacity.

3.1.2 Assumptions of the generic problem

This Section presents significant characteristics of DVRPMB, along with key assumptions.

Regarding the arrival process of DO, we assume that it allows sufficient time for each order to be served by a new vehicle dispatched from the depot prior to the closing of its time window. This assumption secures that there is potential of all DO to be included in the current schedule of the vehicles *en route*, or served by a vehicle located at the depot (or, of course, not be served for the limited-fleet case).

Additionally, we assume the following problem characteristics/assumptions regarding the operating scenarios considered for the DVRPMB:

a) The current status of the logistics operations (i.e. current location of each vehicle of the fleet, availability in terms of remaining capacity and time for service, remaining unserved customers, etc.) is known at any time instance. In practice, this is achieved by employing appropriate fleet monitoring systems.
b) A vehicle commits to travel at the latest possible time. For example, if a vehicle is planned to arrive to a customer prior to the opening of its time window, the vehicle will wait at the location of the previously served customer. This assumption facilitates re-optimization changes in case new orders arrive to the system.
c) 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.

It should be noted that in similar studies, such as the work of Chen and Xu (2006) and the work of Ichoua *et al.* (2000), a time interval δt representing the time needed for the algorithm to run is added to the re-optimization instance T_ℓ and the corresponding solution is then valid for the time period $[T_\ell + \delta t, T_{max}]$ until, of course, the next re-optimization event occurs. In our case we make the simplifying assumption that δt is minimal (practically zero compared with the typical travel time between clients), provided that the computational times of the proposed algorithms are appropriately short.

3.2 Re-optimization in DVRPMB

As mentioned already, the allocation of DO in the available fleet is dealt through iterative *re-optimization* as described below. The related solution strategy needs to define the following basic components:

i. *The re-optimization problem:* i.e. the static optimization problem to be solved at each re-optimization cycle based on the information available.
ii. *The re-optimization cycle:* i.e. the interval between two consecutive re-optimization steps.
iii. *The re-optimization tactic:* i.e. the way of introducing newly received DO to the fleet for service, i.e. release all planned DO immediately for implementation or release for implementation only the DO scheduled for service prior to the next re-optimization cycle (i.e. during the next cycle, re-consider all DO not yet served)[3].

Items (i) and (ii) concern the re-optimization process, or strategy. Below we formulate the re-optimization problem, and in Chapter 4 we propose an exact and a heuristic algorithm to solve it. In Chapters 5 and 6 we study the re-optimization cycle and the re-optimization tactic for the cases of unlimited and limited vehicle fleets, respectively.

3.2.1 Solution framework

We assume that in the overall planning horizon $[0, T_{max}]$, there will be L *re-optimization cycles*, each corresponding to an appropriate "static" problem $\Gamma_1, \Gamma_2, \dots, \Gamma_L$, with *re-optimization* occurring at time instances $T_\ell, \ell = 1,2, \dots, L$ where $T_0 = 0 < T_1 < \cdots < T_L < T_{max} - \tau$. *Re-optimization* cycles ($[T_{\ell-1}, T_\ell], \ell \geq 1$) may not be necessarily of equal duration and may not even be known *a priori* (e.g. when re-optimization depends on the number of DO received – see Chapter 5, Section 5.2). The "*static*" problem solved at each *re-optimization* time T_ℓ, denoted as DVRPMB(ℓ), considers all information known up to the related point in time. It is assumed that this problem (Γ_ℓ) is solved instantaneously. The structure of the *re-optimization* framework is illustrated in Figure 3.1 and described below.

At the beginning of the planning horizon (T_0), there is a set of known (static) orders, and a sufficient number of vehicles located at the depot that may serve all these orders (even in the case of limited fleet). Based on this information, a set of initial routes $R_s = \{r_1, r_2, \dots, r_K\}$ has been developed to serve the related orders. This initial solution S_0 obtained at $t = 0$ is defined over the planning horizon $[0, T_{max}]$.

A re-optimization problem $\Gamma_\ell, \ell \in \{1, \dots, L\}$ takes into account two sets of orders not yet served: i) the *committed orders* that include all orders assigned to a vehicle originally or during previous re-optimization cycles, which have not been served and cannot be re-allocated to other vehicles,

[3] The implementation of this tactic depends on the technology used. Typically the driver receives only the DO to be served next.

and ii) the *flexible orders*, that correspond to newly arrived DO, or previously arrived DO not yet served. Typically, flexible orders correspond to all DO that have not been served during the current re-optimization cycle (time T_ℓ). However, there are some practical cases in which this may not be applicable, and DO assigned to vehicles during a prior re-optimization cycle, may be considered as committed orders. This limitation may be caused by committed financial transactions, prior communications with the customers, etc. For the reason above, depending on the policy, two scenarios are relevant: a) committed orders correspond only to offline requests and flexible orders are all DO not yet served, and b) committed orders are all orders assigned to vehicles during previous re-optimization cycles and not yet served; flexible orders correspond only to newly arrived DO. Those two cases will be analyzed subsequently in Chapter 4.

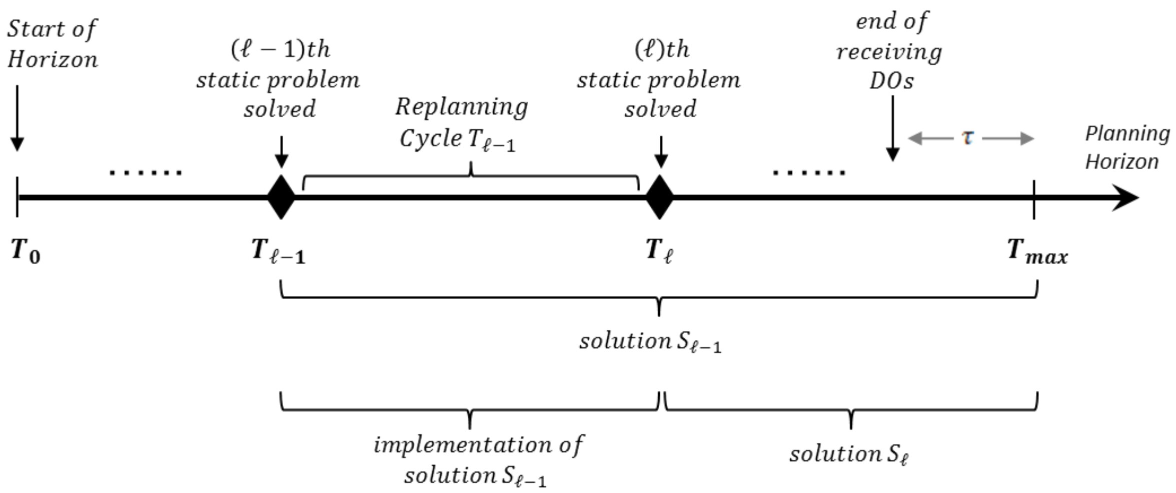


Figure 3.1. Overview of the re-optimization framework

In practice, the solution framework of Figure 3.1 may be implemented using current communication and information technologies. Requests arriving in real-time through a call center are used as inputs into the planning system. The dispatcher chooses a certain re-optimization policy, and an appropriate part of the resulting plan is transmitted to the drivers via onboard devices or PDAs.

The vehicles involved in the static problem of re-optimization cycle ℓ include: i) those that were dispatched earlier (even at time T_0) but have not returned to the depot by time T_ℓ; and ii) the ones located at the depot. Note that the number of the latter is considered sufficiently large for the unlimited fleet case. The solution S_ℓ of the static problem of re-optimization cycle ℓ concerns the entire remaining time horizon $[T_\ell, T_{max}]$. Part of this solution is then implemented until the next re-optimization trigger, i.e. at time $T_{\ell+1}$. This process is illustrated in Figure 3.2.

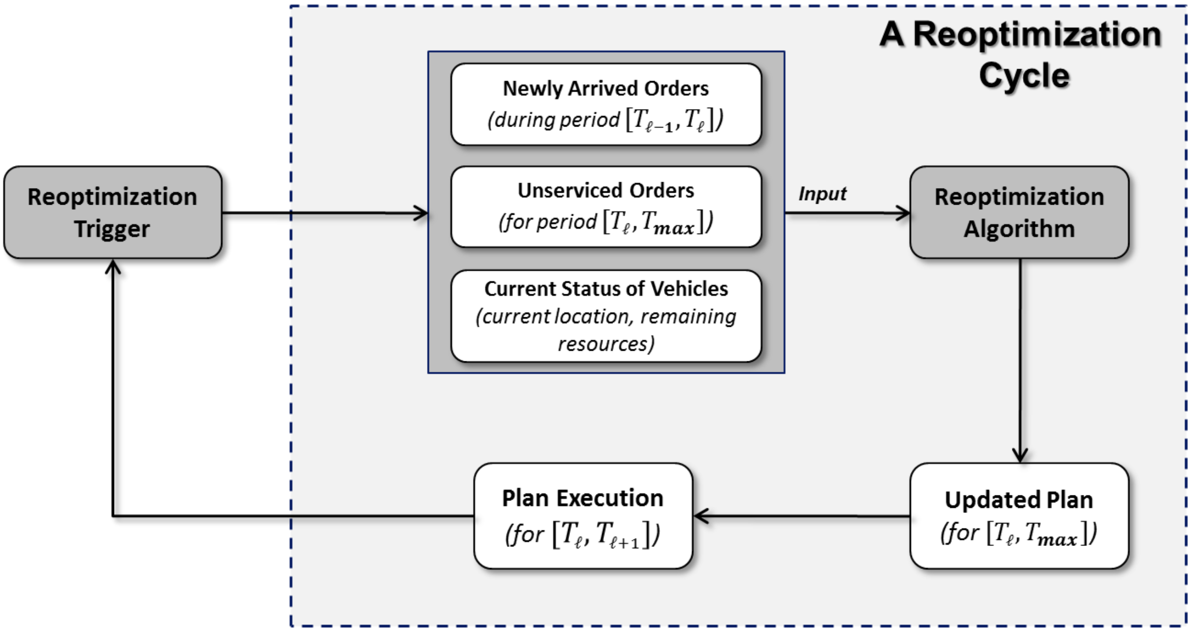


Figure 3.2. A simple re-optimization process

In the following, we specify precisely the sets of orders N_ℓ, and the sets of vehicles V_ℓ involved in the static problem of each re-optimization cycle ℓ, $\ell = 1,2, \dots, L$.

Suppose that we have solved the static problem of re-optimization cycle $(\ell - 1)$ and obtained a solution $S_{\ell-1}$ for some $\ell \in \{2, \dots, L\}$. In order to implement this solution at time $T_{\ell-1}$, a set of vehicles from $V_{\ell-1}$ is used to serve known orders $N_{\ell-1}$ according to solution $S_{\ell-1}$. Denote those vehicles as $K_{\ell-1}$. Obviously, $K_{\ell-1} \subseteq V_{\ell-1}$ and contains a finite number of vehicles. By time T_ℓ (next re-optimization cycle), some of the vehicles, denoted as $K_{\ell-1}^C$, may have completed their trips and returned to the depot, while the others are *en route* and may still have capacity available to serve additional orders. In the static problem of the next re-optimization cycle ℓ, the vehicles in set $K_{\ell-1} \backslash K_{\ell-1}^C$ and the remaining vehicles located at the depot K_ℓ^d can be used. Thus, the set of vehicles involved in the static problem of re-optimization cycle ℓ, is: $V_\ell = (K_{\ell-1} \backslash K_{\ell-1}^C) \cup K_\ell^d$.

Let $N_{\ell-1}^C \subseteq N_{\ell-1} -$ denote the set of orders already served during the implementation of the portion of solution $S_{\ell-1}$ in interval $[T_{\ell-1}, T_\ell]$, and $N_{\ell-1}^d$ the set of new orders received during the same interval. Then, the set of orders to be considered in the static problem of re-optimization cycle ℓ, is: $N_\ell = (N_{\ell-1} \backslash N_{\ell-1}^C) \cup N_{\ell-1}^d$. Figure 3.3 illustrates sets V_ℓ and N_ℓ that form the static problem of each re-optimization cycle ℓ. Based on this, we provide in the next subsection the mathematical formulation of the static problem of each re-optimization cycle.

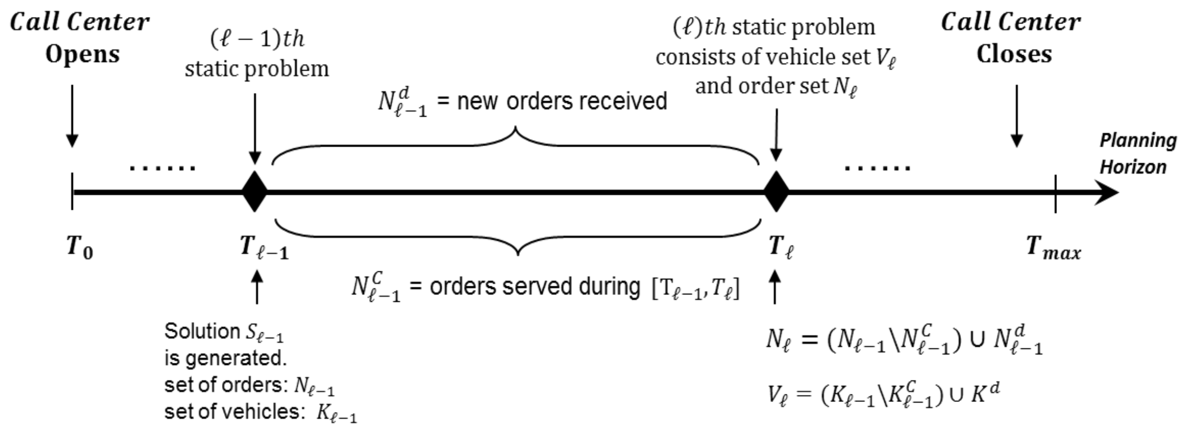


Figure 3.3. Sets of vehicles and orders considered in the static problem

3.2.2 Mathematical formulation of DVRPMB(ℓ)

In describing the re-optimization problem we omit index ℓ, since the problem has the same form for any re-optimization cycle (for example, in the following, time T corresponds to re-optimization trigger time T_ℓ).

Let $N = C \cup F$ denote the set of orders which have not been served, where C and F denote the sets of known committed and flexible orders, respectively. Furthermore, $C = \bigcup_{k\in K} C_k$, where C_k represents the set of committed orders assigned to vehicle k that is *en route*. Note that C_k may include both delivery and pick-up orders that are assigned to vehicle $k \in K$ during previous re-optimization cycles and cannot be re-assigned to other vehicles. Let set $M = \bigcup_{k\in K}\{\mu_k\}$, where μ_k represents the current location of vehicle $k \in K$, and node 0 represent the origin/destination depot. We consider a complete directed graph in a Euclidean plane $G = (W, A)$, where $W = C \cup F \cup M \cup \{0\}$, and A the set of arcs connecting all nodes $W (A = \{(i,j): i \in W, j \in W\backslash M\})$. The cost of traversing arc $(i,j), \{i \in W, j \in W\backslash M\}$ is denoted by c_{ij}, while t_{ij} denotes the travel time between these two nodes (assuming that cost matrix $[c_{ij}]$ satisfies the triangular inequality).

Each order $i \in N$ is related to the following quantities:

d_i	is the demand/supply of the order at each client site (load to be delivered or picked-up by a vehicle). Delivery orders are associated with a negative value and pick-up orders with a positive one. The demand/supply of the depot is zero ($d_0 = 0$).
s_i	is the service time of order i at the client site; $s_0 = 0$
h_i	is the arrival time of a new order i. Obviously, $0 < h_i < T_{max} - \tau$, $\forall i \in F$ and $h_i = 0, \forall i \in C$

$[a_i, b_i]$ is the time window of order i. For orders known prior to time T_0, $0 \leq a_i < b_i \leq T_{max}$ and for DO, $h_i < a_i < b_i \leq T_{max}$. Additionally, $a_0 = 0$ and $b_0 = T_{max}$. The time window of a customer cannot be violated, i.e. order i must be served within this time window.

The proposed mathematical formulation involves three (3) types of decision variables: i) binary flow variables x_{ijk}, equal to 1 if arc $(i,j) \in A$ is traversed by vehicle $k \in V$ and zero otherwise, ii) time variables w_{ik}, which represent the start of service for order $i \in N$ by vehicle $k \in V$, while for the depot $w_{0k} \geq T$, and iii) load variables Q_{ik}, which provide the load of vehicle $k \in V$ immediately after serving node $i \in W$. Note that the initial load $Q_{\mu_k k}$ of vehicle $k \in K$ at each re-optimization cycle is equal to the total amount to be delivered (and/or picked up) by vehicle k (i.e. remaining SO originally assigned to it but not yet served and DO that have been served by the vehicle in the past, but not yet returned to the depot).

The re-optimization model for DVRPMB is similar to the formulation proposed by Parragh *et al.* (2008) for the multi-vehicle PDP, which was, in turn, adapted from the model proposed by Cordeau *et al.* (2002) for the VRPTW.

We first present the model for the unlimited-fleet case. Subsequently, the modifications needed for the limited fleet case are discussed.

Unlimited-fleet Case

The objective of the problem is to minimize the total cumulative routing cost over the planning horizon $[T_\ell, T_{max}]$ and is given by:

$$\min(z) = \sum_{k \in V} \sum_{(i,j) \in A} c_{ijk} x_{ijk} \qquad (\Pi.1)$$

Subject to:

$$\sum_{j \in C_k \cup F \cup \{0\}} x_{ijk} = 1 \qquad \forall k \in K, \forall i \in C_k \cup \{\mu_k\} \qquad (\Pi.2$$

$$\sum_{k \in V} \sum_{j \in W} x_{ijk} = 1 \qquad \forall i \in F \qquad (\Pi.3$$

$$\sum_{i \in C_k \cup F \cup \{\mu_k\}} x_{i0k} = 1 \qquad \forall k \in K \qquad (\Pi.4$$

$$\sum_{j \in F} x_{0jk} \leq 1 \qquad \forall k \in K^d \qquad \text{(Π.5}$$

$$\sum_{j \in F} x_{0jk} = \sum_{j \in F} x_{j0k} \qquad \forall k \in K^d \qquad \text{(Π.6}$$

$$\sum_{i \in W} x_{ihk} - \sum_{j \in W} x_{hjk} = 0 \qquad \forall h \in N, \forall k \in V \qquad \text{(Π.7}$$

$$Q_{jk} \geq Q_{ik} + d_j - Z(1 - x_{ijk}) \qquad \forall (i,j) \in A, \forall k \in V \qquad \text{(Π.8}$$

$$\max\{0, d_i\} \leq Q_{ik} \leq \min\{\bar{Q}, \bar{Q} + d_i\} \qquad \forall i \in N, \forall k \in V \qquad \text{(Π.9}$$

$$w_{jk} \geq w_{ik} + s_i + t_{ij} - Z(1 - x_{ijk}) \qquad \forall (i,j) \in A, \forall k \in V \qquad \text{(Π.1}$$

$$\max(a_i, T) \sum_{j \in W} x_{ijk} \leq w_{ik} \leq b_i \sum_{j \in W} x_{ijk} \qquad \forall k \in V, \forall i \in W \qquad \text{(Π.1}$$

$$T \leq w_{0k} \leq b_0 \qquad \forall k \in K^d \qquad \text{(Π.1}$$

$$x_{ijk} \in \{0,1\} \qquad \forall (i,j) \in A, \forall k \in V \qquad \text{(Π.1}$$

As mentioned before, the objective function (3.1) expresses the total routing cost over the entire available horizon. Constraint (3.2) specifies that each vehicle k *en route* must serve all committed orders originally assigned to it (including its corresponding starting location). Constraint (3.3) ensures that all flexible orders will be served, either by a vehicle *en route* or by a vehicle available at the depot. Consequently, the above two Constraints ensure that all orders in the system will be served exactly once. Constraints (3.4) force active vehicles *en route* to eventually return to the depot. According to Constraint (3.5) new vehicles dispatched from the depot in the current re-optimization cycle can only serve DO. Constraints (3.6) force these new vehicles to return to the depot. Note also that Constraints (3.5) allow vehicles to remain at the depot if necessary (not all vehicles available at the depot must be used). Constraint (3.7) ensures flow conservation, and Constraints (3.8) and (3.9) ensure that the vehicle's capacity limit is respected at all vertices, where Z is a large positive constant. Constraints (3.10) – (3.11) ensure that a route is time feasible; Constraint (3.10) updates the start time (of service) along the route, while (3.11) ensures that the service start time is within the time window of the node. Note that Z represents a large number, which should be larger than $Z_{ij} = \max(b_i + t_{ij} - a_j, 0)$ for each arc (i,j). Constraints (3.12) force new vehicles K^d to assume duty after the re-optimization time instance and return to the depot within the available planning horizon. Finally, Constraints (3.13) force the flow variables to assume binary values $\{0, 1\}$.

Limited-fleet Case

Under the limited fleet setting, it is possible that not all DO are served either by vehicles *en route* or by vehicles located at the depot. Due to this fact, certain modifications are necessary to the aforementioned generic formulation.

The first modification concerns the customer service constraints, since it is not guaranteed that all DO may be served. Thus, we can relax Constraints (3.3) in as in Constraints (3.14):

$$\sum_{k \in V} \sum_{j \in W} x_{ijk} \leq 1 \qquad \forall i \in F \qquad \text{(П.1}$$

The second modification concerns the objective function of (3.1). Minimizing routing cost is no longer an appropriate objective, since it would preclude service of any dynamic (pick-up) orders. A more appropriate and conventional objective would be to optimize a functional that takes into account the number of dynamic orders served, and the routing cost. This objective is modeled by the modified functional:

$$\min(z) = -\xi_u \sum_{k \in V} \sum_{(i,j) \in A | i \in F, j \in W} x_{ijk} + \sum_{k \in V} \sum_{(i,j) \in A} c_{ijk} x_{ijk} \qquad \text{(П.1}$$

where ξ_u is a profit assigned for each DO served. If lexicographical (i.e. service is prioritized over routing costs), then the profit for serving a DO should be higher that the routing costs for incorporating this DO in the plan; if not, then the solution will not include this DO, since the overall objective will increase. Thus, ξ_u may be larger than $\max_{i \in F}(c_{r_i})$, where c_{r_i} represents the cost of the unit route $[Depot - i - Depot]$. Note that the suitability of such objective in a dynamic problem such the one in hand is discussed in detail in Chapter 6. More suitable objective functions are also proposed there.

Note on complexity

The DVRPMB(ℓ) is NP-hard in the strong sense, since it generalizes the basic version of VRPTW, arising when $\{\mu_k\} = \{0\}, \forall k \in K$, which has been proven to be NP-hard in the strong sense (Toth and Vigo, 2002). This can be also supported by the following considerations:

a) In case there is only one vehicle involved, i.e. $k = 1$, the DVRPMB(ℓ) generalizes the well-known TSPTW, arising when $d_i = |d_i|, \forall i \in F$, which is a special case of the VRPTW.

b) In case there are more than one vehicles involved, i.e. $k > 1$, the DVRPMB(ℓ) generalizes the VRPTW, arising when $C_k = \emptyset, \forall k \in K$ and $d_i = |d_i|, \forall i \in F$.

To illustrate the computational implications of DVRPMB(ℓ)'s complexity, consider Figure 3.4. The latter concerns the case in which a simple exhaustive search algorithm is applied, which examines all available DO to be incorporated to all feasible insertion places within the existing routes. If no feasible location exists for a DO, a new vehicle will be dispatched from the depot to serve this order. In this, simpler than the one examined in the current Chapter case (due to fixed sequence of delivery orders), the Figure shows that an exhaustive algorithm can be computationally intractable for cases in which the number of DO is higher than say 8 or 9. These illustrative results were obtained considering a set of 20 SO, 2 vehicles and 1 to 8 DO[4].

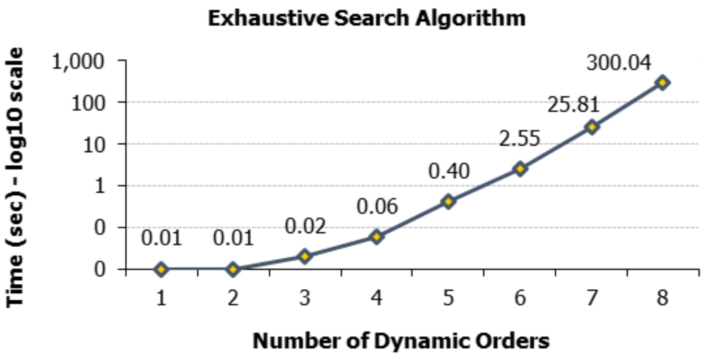


Figure 3.4. Computational time increases prohibitively as number of requests increase

[4] The exhaustive algorithm was implemented on MATLAB® 2009 and solved in a Dual-Core Windows 7 machine with 2GHz processors and 2GB RAM.

Chapter 4: BRANCH-AND-PRICE ALGORITHM FOR THE RE-OPTIMIZATION PROBLEM

This Chapter presents the solution approach for the re-optimization problem of DVRPMB and for the case of unlimited fleet. This problem is solved at each re-optimization cycle T_ℓ, as described in Chapter 3. In Chapter 6 we consider the re-optimization problem for the case of limited fleet.

Section 4.1 provides an overview of the branch-and-price (B&P) method and how it is applied to the problem in hand. Section 4.2 formulates the re-optimization problem in a set-partitioning model and discusses the initial feasible solution, which is provided as input to the column generation algorithm. Section 4.3 presents the framework that identifies variables (columns) to be added to the initial set that can further enhance the objective value (pricing sub-problem). Section 4.4 discusses the solution mechanism for the pricing sub-problem to obtain optimal solutions, which corresponds to an Elementary Shortest Path Problem with Resource Constraints (ESPPRC). Section 4.5 presents a conceptual synthesis of the overall column generation method for the re-optimization problem of DVRPMB, while Section 4.6 discusses the proposed Branch-and-Price algorithm to obtain optimal integer solutions. Finally, Section 4.7 proposes a heuristic-based approach for the pricing sub-problem that produces near-optimal solutions for practical cases with extended solution space for which the optimal approach may not return a solution within reasonable computational times (e.g. cases without time windows).

4.1. Overview of the B&P approach

We propose a new *branch-and-price (B&P)* approach to solve the re-optimization problem of DVRPMB. The B&P algorithm consists of a column generation (CG) algorithm embedded within a branch-and-bound (B&B) scheme; CG is used to compute lower bounds at each node of the B&B search tree, while B&B is used to obtain the optimal integer solution.

In this CG framework, the formulation presented in Chapter 3 is decomposed to a Master Problem (MP) and to several Sub-problems (SP). For formulating the MP we employ a set partitioning model, which is typically used in Column Generation formulations of the VRP. In the set partitioning formulation, each column corresponds to a feasible route and each constraint corresponds to a customer. Consequently, MP involves only constraints that impose a single visit to each customer. All other constraints are handled in the sub-problems. A detailed description of the set-partitioning formulation is presented in Section 4.2.1.

Only a portion of known feasible routes are used to form the Restricted Master Problem (RMP). To preserve feasibility in the RMP, low quality columns (routes) are oftentimes used in the initial set of feasible columns (e.g. single-visit routes, i.e. $depot - order\ i - depot$). Obviously, better quality columns lead to faster convergence. For that reason, in the dynamic problem setting of DVRPMB we exploit the information of previous re-optimization cycles. By modifying appropriately the columns corresponding to a feasible solution of the previous re-optimization cycle (as described in detail in Section 4.2.2), and adding new columns corresponding to the newly received orders, we provide an initial set of feasible columns and solve a linear relaxation[5] of the re-optimization problem in the current cycle.

In order to identify variables (columns) that have a negative reduced cost w.r.t. the dual solution of the RMP, a different optimization problem is solved (sub-problem), called the pricing problem. This latter problem handles all remaining constraints that a column (route) is required to satisfy. Such constraints include the requirement for serving the committed orders assigned to each vehicle, as well as all resource constraints. We propose both an exact and a heuristic approach to solve this problem. For the former case we formulate the pricing problem as an Elementary Shortest Path Problem with Time Windows and Capacity Constraints (ESPPTWCC), and we employ a dynamic programming-based method to solve it. For the latter case, we employ an insertion-based heuristic that uses the information of the dual prices

[5] The RMP is solved by relaxing the variables that denote if a column (route) is included in the solution to assume fractional values (instead of binary).

returned by the solution of the linear relaxation of the RMP. In both cases, the solution is one or more columns/routes that minimize a certain objective function. If the solution of the pricing problem is non-negative, then the solution of the linear relaxation of RMP is optimal; otherwise, the resulting column(s) may enter the basis and is (are) added to the current collection of columns of the RMP.

The proposed *column generation scheme* for the re-optimization problem comprises the following steps:

Step 1. Restricted Master Problem (RMP)

Generate an initial set of columns comprising a feasible solution to the RMP: To do so, modify the solution (routes) from the previous re-optimization cycle (in order to represent the up-to-date information) and add single-visit columns for flexible (dynamic) orders (*Section 4.2.2*).

Step 2. Solving the linear programming relaxation of RMP

Solve the linear relaxation of the resulting RMP and obtain optimal primal and dual variables (also *Section 4.2.2*).

Step 3. Pricing Problem (sub-problem)

Solve the column generating sub-problem (pricing problem), i.e. identify columns that, if included in the basis of the RMP, they further reduce the objective function value:

- Exact solution: Decompose the complete problem to $|K|+1$ independent sub-problems, where $|K|$ is the number of vehicles *en route*. Each sub-problem is an $ESPPTWCC$, which is solved by the label correcting algorithm. For the K sub-problems, consider orders $C_k \cup F, k \in \{1,2,\dots,K\}$, where C_k and F correspond to the set of committed and flexible orders, respectively. These sub-problems will return columns for each vehicle *en route* that will contain all committed orders C_k and incorporate flexible ones from the F set. For the $K+1$ sub-problem consider only orders F; the returned columns will represent route(s) of newly dispatched vehicles from the depot. (see *Sections 4.3 – 4.4*)
- Heuristic solution: Considering the solution (routes) of the previous re-optimization cycle, generate new columns for vehicles already *en route* that incorporate flexible orders using an insertion heuristic based on dual-prices. Use a limited (heuristic) version of ESPPTWCC to generate columns corresponding to vehicles dispatched from the depot in order to serve only flexible orders (*Section 4.7*).

Step 4. Combining the RMP with the sub-problems

If there is at least one generated column with negative reduced cost corresponding to either a vehicle *en route* or a vehicle located at the depot, add it to the RMP and go to Step 2. If no new negative reduced cost column is found, then stop; the optimal solution (lower bound) has been obtained (see *Section 4.5*).

Note that since the column generation procedure described above operates on the relaxed RMP, 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 described in detail in Section 4.6.

The contribution of our solution methodology compared to typical B&P applications in VRP is three-fold. First, we have introduced an appropriate structure that exploits the characteristics of the dynamic problem in hand. Secondly, we appropriately enhanced the dominance criteria in the solution of the sub-problem in order to discard a number of non-promising paths. Finally, we employed a new heuristic approach to generate new routes; this heuristic may address practical cases with extended solution space.

4.2. A Set-Partitioning formulation for the proposed Master Problem

4.2.1. The Master Problem

As mentioned above from the formulation of Chapter 3 (Section 3.2.2), the Master Problem (MP) incorporates only those Constraints that cannot be treated independently by the pricing sub-problems, i.e. the linking Constraints (3.3). The MP is a *Set Partitioning Problem* (*SPP*), since every customer should be serviced exactly once.

As mentioned in Chapter 3, the re-optimization problem seeks a solution that serves all known orders $N = (\bigcup_{k \in K} C_k) \cup F$[6] during the interval $[T_\ell, T_{max}]$. Under the set-partitioning formulation, the feasible solution space comprises the ***entire set of feasible*** single-vehicle columns (routes), denoted as Ω. The latter comprises two separate sub-sets, $\Omega = (\bigcup_{k \in K} \Omega_k) \cup \Omega_p$, where:

i. The columns in sets Ω_k correspond to vehicles K already *en route*; each one of those columns/routes should originate from a current vehicle location μ_k, end at the depot, and include all committed to this vehicle orders (C_k) and, perhaps, some flexible orders.

[6] Note that for simplicity purposes, we assume that set C_k comprises all unserved committed orders of vehicle k and set F contains all unserved flexible (dynamic) orders; those sets change in every re-optimization cycle as already stated in Chapter 3.

ii. Columns in Ω_p correspond to vehicles K^d located at the depot. These routes originate and end at the depot, and include only F orders.

All above routes originate at time T_ℓ, and end when the corresponding vehicle returns to the depot (the latest at T_{max}). Furthermore, for each route, all resource constraints must be satisfied.

It should be noted that set F comprises the following:

- DO that have arrived during previous re-optimization cycles (i.e. prior to time $T_{\ell-1}$), but not yet served, and
- DO that have been received during the interval $[T_{\ell-1}, T_\ell]$.

Although orders of the first category above have been assigned to some vehicles during the solution $S_{\ell-1}$ of the previous re-optimization problem (if any), we treat those orders as flexible and allow them to be served by any vehicle $k \in V$ during re-optimization cycle ℓ. If in $S_{\ell-1}$ a new vehicle k has been dispatched to serve F orders (now considered as vehicle *en route* with $C_k = \emptyset$), all DO assigned to it and not yet served are also considered as flexible orders. In Chapter 5, we also consider a policy for which any DO assigned to a certain vehicle during solution $S_{\ell-1}$ is restricted to be served by that vehicle only. However, the solution approach remains similar. Thus in the rest of the current Chapter we present the approach referring to the first problem variation.

In order to formulate the **Master Problem**, we introduce binary coefficients e_{ir} and y_r, such that:

$$e_{ir} = \begin{cases} 1, & \text{if order } i \in N \text{ is included in route } r \in \Omega \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

$$y_r = \begin{cases} 1, & \text{if route } r \in \Omega \text{ is used in the solution} \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

If c_r denotes the cost of route $r \in \Omega$, then the formulation of the Master Problem is as follows:

$$(\boldsymbol{SPP}) \quad Minimize \quad \sum_{r \in \Omega} c_r y_r \tag{4.3}$$

$$subject\ to: \quad \sum_{r \in \Omega} e_{ir} y_r = 1 \qquad \forall i \in N \tag{4.4}$$

$$y_r = \{0, 1\} \qquad \forall r \in \Omega \tag{4.5}$$

The above formulation seeks to find a subset of single vehicle routes in Ω that minimizes the total distance and serves each order in N exactly once [Constraint (4.4)]. We denote this formulation by SPP. Eliminating binary Constraints (4.5) (or relaxing them to $y_r \geq 0$), permits the problem to be solved using known linear programming techniques. We will refer to the resulting linear relaxation problem as $LP - SPP$.

Note that our formulation ensures that exactly one route in set Ω_k (i.e. for each vehicle *en route*) will participate in the optimal solution. Consider a case of K vehicles *en route*, each of which is located at a certain location μ_k and is tasked to serve a set $C_k, k \in K$ of unserved committed orders. Since all routes in Ω are feasible (as guaranteed by the solution of the sub-problems presented in Section 4.3), column Ω_k includes all committed orders C_k as well as the initial vehicle location μ_k, since $e_{ir} = 1, \forall i \in (C_k \cup \mu_k), \forall r \in \Omega_k, k \in K$ (Constraint 3.4 of Chapter 3). Since all columns are feasible, partitioning Constraints (4.4) assign each column in the set Ω_k to at most one vehicle (each vehicle $k \in K$ will be used at most once).

However, flexible orders (F set) can be assigned to vehicles *en route* (K set) or to vehicles located at the depot (K^d set). For the former case, a column, which includes orders C_k and the origin μ_k, could also contain flexible orders. The latter can be formulated as a typical VRPTW set-partitioning problem.

4.2.2. The Restricted Master Problem, RMP

As already mentioned, the MP formulation requires the explicit enumeration of all columns *a priori*. Even if all feasible columns could be somehow found, the LP − SPP could not be solved within reasonable computational time.

Suppose that a subset $\Omega' \subset \Omega$ of feasible routes is known and forms the basis for the RMP. Based on this restricted set, we may define a restricted version of SPP, denoted as $\boldsymbol{SPP^*}$. Consider now the following linear programming relaxation of the RMP (denoted as $\boldsymbol{LP - SPP^*}$) involving the set Ω':

$$(\boldsymbol{LP - SPP^*}) \quad Minimize \quad \sum_{r \in \Omega'} c_r y_r \tag{4.6}$$

$$subject\ to: \quad \sum_{r \in \Omega'} e_{ir} y_r = 1 \qquad \forall i \in N \tag{4.7}$$

$$y_r \geq 0 \qquad \forall r \in \Omega' \tag{4.8}$$

Finding the initial set of columns Ω' that contain a feasible solution is not a trivial task. In order to construct this set, we exploit the information from solution $S_{\ell-1}$ obtained during the re-optimization cycle $(\ell - 1)$ for the interval $[T_{\ell-1}, T_{max}]$. Eliminating all orders that have been served up to T_ℓ (i.e. during $[T_{\ell-1}, T_\ell]$) yields a feasible solution $S'_{\ell-1}$ of routes that comprise two types of columns corresponding to vehicles *en route*:

- Those dispatched at time $T_0 = 0$, which should serve remaining committed orders
- Those dispatched from the depot at time $T_{\ell'}$, where $0 < \ell' \leq \ell - 1$, which serve DO arrived during previous re-optimization cycles

Following this process, a feasible set of columns (routes) is generated and used as an initial set Ω' in the corresponding $LP - SPP^*$. A note here about committed orders: In addition to the static orders assigned to vehicles prior to the start of operations, committed orders may include DO (received during previous re-optimization cycles and not yet served) depending on the policy followed (see Chapter 3 – Section 3.2).

This (feasible) set of routes of vehicles *en route* (set K) that cover all committed orders may be used as an initial solution in the set Ω' of the corresponding RMP in order to represent columns Ω_k in the restricted column-set Ω'. For the flexible orders (F set), we generate single-visit trips that originate and finish at the depot, i.e. $[depot - i - depot], \forall i \in F$ to be added to the initial set of columns Ω' of the RMP (Ω_p columns).

A technical implementation issue that is worth mentioning corresponds to degeneracy issues[7] caused by potential redundant constraints (rows), even if the initial set of columns Ω' comprises a feasible set of routes. In order to avoid degeneracy issues, we can add to this former set, single-visit trips (columns) for all committed orders of each route in K. Since it is not desirable to include such columns in the final solution of $LP - SPP^*$ (i.e. a new vehicle to be dispatched from the depot in order to a serve committed order), we incorporate the columns in the initial basis with a sufficiently large cost c_r.

The RMP can be solved using known linear programming techniques (e.g. *Simplex* or the *Revised Simplex Method*). The solution also generates the dual (shadow) prices, which are provided to the pricing problem in step 3 and used in order to compute the reduced costs.

[7] An LP is degenerate if in a basic feasible solution one of the basic variables assumes a value of zero. Degeneracy is caused by redundant constraint(s) and could necessitate additional iterations in Simplex.

Suppose that $LP-SPP^{*}$ (RMP) has a feasible solution, and let π be the associated dual solution, i.e. the dual variables π_i $(i \in N)$ that are associated with partitioning Constraints (4.7). Note that the solution provided by the current RMP is optimal with respect to the columns (routes) of the Ω' set. In order to check if this solution is globally optimal for the MP, we should calculate the reduced costs ($\widehat{c_r}$) of each non-basic route $r \in \Omega$. According to the duality theory of linear programming, the solution is optimal with respect to $LP-SPP$ (MP) if and only if the reduced cost $\widehat{c_r}$, is nonnegative for each r in the global set Ω, i.e.,

$$\widehat{c_r} = c_r - \sum_{i \in N} e_{ir}\pi_i \geq 0 \qquad \forall r \in \Omega \tag{4.9}$$

In order to provide a test for the optimality of this solution with respect to $LP-SPP$ (MP), i.e. to check whether there exist negative reduced cost variables, one could solve the following minimization problem ***(pricing problem)***:

$$min\{c_r - \textstyle\sum_{i \in N} e_{ir}\pi_i \mid r \in \Omega\} \tag{4.10}$$

Let z denote the value of the solution to the pricing problem in (4.10), and let r_z denote the corresponding route. If $z \geq 0$, then x is also optimal with respect to $LP-SPP$; otherwise r_z defines a column that can enter the basis and should be added to Ω'. In this case the $LP-SPP^{*}$ is solved again. This iterative procedure continues until no other negative columns exist.

4.3. The Pricing Sub-Problem

Having solved the RMP (by known linear programming techniques), a pricing sub-problem (*SP*) is solved to identify variables (columns) in $\Omega \backslash \Omega'$ with negative reduced cost w.r.t. the dual solution of the RMP (Desaulniers *et al.*, 2005).

In order to address the requirement that committed orders cannot be re-distributed among vehicles, we formulate and solve several independent SPs, one for each vehicle *en route* (K set). We denote these independent SPs by Ψ_k, $\forall k \in K$. The set of orders considered for each Ψ_k consists of the remaining committed orders of vehicle k (C_k set) plus all F orders, i.e. $N_k = C_k \cup F$. This F set is common in each Ψ_k. The solution of each Ψ_k will generate feasible trips (columns) that originate from the current vehicle location μ_k and cover all remaining C_k orders and potentially some orders from the F set. The subset of columns generated by each Ψ_k will comprise set $\Omega_k, k \in K$.

In order to consider also the assignment of F orders to vehicles located at the depot, we solve an additional independent SP, denoted as $\Psi_{|K|+1}$, that includes only the F set, i.e. $N_{|K|+1} = F$. The solution of this problem generates feasible trips (subject to all constraints) that originate from the depot, serve one or more F orders and return to the depot. The columns generated from $\Psi_{|K|+1}$ comprise set Ω_p. Figure 4.1 illustrates the proposed decomposition approach.

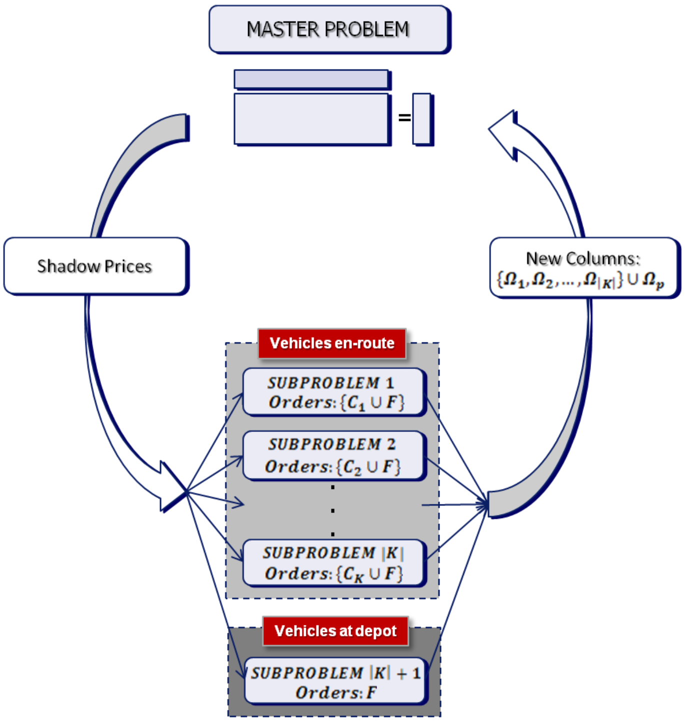


Figure 4.1. Illustration of the decomposition approach for the pricing sub-problem

Since the $|K| + 1$ independent SPs are of a much smaller scale, the aforementioned straightforward approach leads to an efficient generation of a large set of feasible columns to be added to the basis. The proof that the proposed decomposition approach provides the optimal solution is relatively straightforward.

Claim: *Given the above notation, the set of columns to enter the basis when solving the* $|K| + 1$ *independent SPs (corresponding to order sets* C_k *and* F*), is exactly the same to the one provided by the solution of a single monolithic sub-problem.*

Proof: As mentioned previously, the monolithic pricing sub-problem seeks to find the route with the minimum cost among the set of columns not yet examined. This set, denoted by $\Omega'' = \Omega \backslash \Omega'$, is given by:

$$z = min\{c_r - \sum_{i \in N} e_{ir} \pi_i \mid r \in \Omega''\}$$

For the proposed decomposed approach, let $\Omega_k'' \coloneqq \{\Omega_1, \Omega_2, \dots, \Omega_{|K|}\}$ be the columns corresponding to the solution of Ψ_k, and Ω_p'' be the ones corresponding to the solution of problem $\Psi_{|K|+1}$. Then, the optimization problem that involves the different independent SPs can be formulated as:

$$z = \min\left(\min_{k\in K}\left\{\min\{c_r^k - \sum_{i\in\{C_k\cup F\}} e_{ir}\pi_i \mid r \in \Omega_k''\}\right\}, \min\{c_r - \sum_{i\in F} e_{ir}\pi_i \mid r \in \Omega_p''\}\right)$$

Since only feasible columns are involved in the Ω_k'' set, the minimization problem for each $k \in K$ can be eliminated, and thus, the objective may assume the following form:

$$z = \min\left(\min\{c_r - \sum_{i\in\{C_k\cup F\}} e_{ir}\pi_i \mid r \in \Omega_K''\}, \min\{c_r - \sum_{i\in F} e_{ir}\pi_i \mid r \in \Omega_p''\}\right)$$

Considering that $\Omega'' \coloneqq \{\Omega_1, \Omega_2, \dots, \Omega_{|K|}\} \cup \Omega_p''$ and $N \coloneqq \{C_1, C_2, \dots, C_k\} \cup F$, the problem is also equivalent to:

$$z = \min\{c_r - \sum_{i\in N} e_{ir}\pi_i \mid r \in \Omega''\},$$

which is exactly the same as the one of the monolithic approach.

Note that sub-problems $\Psi_k, k \in K$ and $\Psi_{|K|+1}$ are solved using the same approach, that is, the as an Elementary Shortest Path Problem with Resource Constraints (ESPPRC), based on the work of Irnich and Desaulniers (2005).

As already discussed, in order to check if this solution is globally optimal for the MP, we should calculate the reduced costs of each non-basic route $r \in \Omega\backslash\Omega'$. Assuming node μ' as the source point (which may represent either the depot 0 or the current location μ_k), the reduced cost $\hat{c}_r$ of path r from μ' to the depot is given by the following equation:

$$\widehat{c_r} = \sum_{(i,j)\in A_r} (c_{ij} - \pi_i^u) \qquad \forall r \in \Omega'' \qquad (4.11)$$

where A_r is the set of arcs in the corresponding path, π_i^u is the value of the dual variable in the dual solution of the RMP at iteration u (of the CG algorithm), and $\pi_{\mu'}^u = 0$. The calculation of Eq. (4.11) for every route contained in the current RMP is straightforward, since all elements are known. By replacing all arc costs c_{ij}, $(i,j) \in A$ by cost factors c_{ij}', the cost of a (feasible) route $r \in \Omega''$ becomes the reduced cost of this route. Therefore, the next step is to generate routes $\grave{r} \in \{\Omega\backslash\Omega'\}$ that have not yet been included in the current RMP, along with their reduced costs $\widehat{c_{\grave{r}}}$. To do so, we solve $|K| + 1$ sub-problems, as previously discussed, and for each sub-problem the route $\grave{r}^*$ with the minimum reduced cost is derived as shown in Eq. (4.12).

$$\widehat{c_{\dot{r}^*}} = \min_{\dot{r}} \left(\sum_{i \in N} e_{i\dot{r}} c'_{ij} \right) \tag{4.12}$$

The modified costs c'_{ij} of each arc $(i,j) \in A$, which may also be negative, are given by the following Equation:

$$c'_{ij} = \begin{cases} c_{ij} - \pi_i^u, & \forall i \in N, j \neq \{\{\mu'\} \cup \{0\}\} \\ c_{ij}, & i = \mu', j \neq 0 \\ +\infty, & i = 0 \\ +\infty, & j = \{\{\mu'\} \cup \{0\}\} \end{cases} \tag{4.13}$$

The scope of each sub-problem is to define the values of coefficients $a_{i\dot{r}}$ that minimize the related reduced cost. Thus, in order to formulate the ESPPRCTW sub-problem we substitute coefficients $a_{i\dot{r}}$ by binary arc flow variables x_{ij} and Eq. (4.12) can be written as follows:

$$min \sum_{(i,j) \in A} c'_{ij} x_{ij} \tag{4.14}$$

As discussed above, the solution of (4.14) should be restricted to generate only feasible routes. Thus, the problem is solved by respecting Constraints (4.15) – (4.22) [which are related to Constraints (3.4) – (3.12) of Chapter 3). Note that subscript k denoting the vehicle is dropped from this formulation, since the vehicles are identical and the relevant vehicle constraints remain in the RMP. Thus, the model constraints are the following:

$$\sum_{i \in N \cup \{\mu'\}} x_{i0} = 1 \tag{4.15}$$

$$\sum_{j \in N \cup \{0\}} x_{\mu' j} = 1 \tag{4.16}$$

$$\sum_{i \in N \cup \{\mu'\}} x_{ih} - \sum_{j \in N \cup \{0\}} x_{hj} = 0 \qquad \forall h \in N \tag{4.17}$$

$$Q_j \geq Q_i + d_j - Z(1 - x_{ij}) \qquad \forall (i,j) \in A \tag{4.18}$$

$$max\{0, d_i\} \leq Q_i \leq min\{\bar{Q}, \bar{Q} + d_i\} \qquad \forall i \in N \tag{4.19}$$

$$w_j \geq w_i + s_i + t_{ij} - Z(1 - x_{ij}) \qquad \forall (i,j) \in A \tag{4.20}$$

$$max(a_i, T) \sum_{j \in N \cup \{\mu'\} \cup \{0\}} x_{ij} \leq w_i \leq b_i \sum_{j \in N \cup \{\mu'\} \cup \{0\}} x_{ij} \qquad \forall i \in N \cup \{\mu'\} \cup \{0\} \tag{4.21}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{4.22}$$

Objective function (4.14) minimizes the sum of the arc reduced costs. Constraints (4.15) to (4.17) ensure that each route starts at the source point and ends at the depot, always preserving the flow along the arcs of the route. Constraints (4.18) and (4.19) ensure that the capacity of the vehicle assigned to the route is not exceeded. Constraints (4.20) and (4.21) ensure that every customer will be served within its time window. Finally, Constraints (4.22) force the flow variables to assume binary values [0,1].

4.4. Solution procedure for the pricing sub-problem

To solve the pricing sub-problems we use a *label correcting algorithm* similar to the one proposed by Feillet *et al.* (2004; 2005). This relies on the creation of multi-dimensional *labels* while processing nodes in an iterative manner. Each label is a vector that corresponds to a partial path δ from the source μ' to vertex $i \in N$, and comprises several components that describe the state of δ, typically the accumulated reduced cost $\tilde{c}_{\delta i}$, as well as the values of the resources upon reaching vertex i as described below.

Typically, in related studies, the corresponding label $[\tilde{c}_{\delta i}, t_{\delta i}, d_{\delta i}]$ represents the accumulated reduced cost, time and demand between the origin and ending node (i) of partial path δ (Irnich and Desaulniers, 2005). In this study, we have introduced a new label component, the *equilibrium cost* $\bar{\bar{c}}_{\delta i}$, which represents an upper bound (worst case) of the total modified cost required to serve all *committed* orders not yet included in partial path δ. Let $O(\delta_i) \subset C$, denote the set of committed orders included in partial path δ ending at vertex i and $O'(\delta_i) \subset \{C \cup F\}$ denote the remaining set of all orders $N = C \cup F$ not yet served by partial path δ. Then, the equilibrium cost can be defined as:

$$\bar{\bar{c}}_{\delta i} = \sum_{i \in C \setminus O(\delta_i)} \left(\max_{h \in O'(\delta_i) \cup \{\mu'\}} (c'_{hi}) + \max_{j \in O'(\delta_i) \cup \{0\}} (c'_{ij}) \right) \tag{4.23}$$

where c'_{ij} is the modified cost associated with arc $(i,j) \in A$. By including $\bar{\bar{c}}_{\delta i}$, the label becomes $\Lambda_{\delta i} = [\tilde{c}_{\delta i}, t_{\delta i}, d_{\delta i}, \bar{\bar{c}}_{\delta i}]$, and indicates whether this partial path δ includes all the required committed orders or not. The label's information is used in the dominance criteria described below.

The procedure commences at the source point μ' with initial label $\Lambda_{\mu'}$ at time $t = t^{\ell}_{\mu'}$. For the $\Psi_{|K|+1}$ sub-problems, $t^{\ell}_{\mu'} = T_{\ell}$; for $\Psi_k, \forall k \in K$, however, a vehicle may be on its way to the next destination or already serving a customer at re-optimization time T_{ℓ}. Therefore, 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.

From source point μ', each label $\Lambda_{\delta i}$ is extended along all arcs $(i,j) \in A$ to create new labels $\Lambda_{\delta' j}$. When extending label $\Lambda_{\delta i} = [\tilde{c}_{\delta i}, t_{\delta i}, d_{\delta i}, \bar{\bar{c}}_{\delta i}]$ to a node j, then the new label $\Lambda_{\delta' j}$ of partial path δ' ending at node j is given be the following extension Equations (note that component $\bar{\bar{c}}_{\delta' j}$ is calculated afresh during each extension according to Eq. (4.23)):

$$\tilde{c}_{\delta' j} = \tilde{c}_{\delta' i} + c'_{ij} \tag{4.24}$$

$$t_{\delta' j} = \max\left\{a_j, t_{\delta' i} + t_{ij} + s_i\right\} \tag{4.25}$$

$$d_{\delta' j} = d_{\delta' i} + d_j \tag{4.26}$$

A label $\Lambda_{\delta' j}$ is discarded if it is not feasible, i.e. if $t_{\delta' j} > b_j$ or $d_{\delta' j} > Q$. Labels are extended based on a procedure which scans all nodes iteratively; each label is extended to all other nodes and checked for feasibility. All new created labels for node j are characterized as non-processed and are stored in a set of non-processed labels, $B(j)$, which is called the bucket of node j. When label $\Lambda_{\delta i}$ has already been extended to all reachable nodes, then it is considered as processed and can be deleted (or kept for supporting the dominance criteria, as will be described later). This is repeated for every $B(j)$ in an iterative manner until all labels have been processed. Following the work of Chabrier (2006), we also adopt the concept of storing all labels that have been extended to all successors in the set of processed labels $\check{P}(j)$, separately for each node j. The adoption of this concept supports the solution process during the dominance checks, as will be described below.

When a partial path is extended to the ending node 0, then a full feasible path has been generated. This path is a potential solution to the minimization problem. For our case, all labels created for ending node 0 are directly stored if and only if they satisfy the following conditions: i) the criterion of negative reduced cost, i.e. $\tilde{c}_{\delta i} < 0$, and ii) all C orders are included in the solution.

During label generation, it is required to consider the constraint of not revisiting the same vertex, i.e. to extend labels strictly to nodes that have not yet been visited (elementary paths, Feillet *et al.*, 2004). To do so, we include an additional component in the label, denoted as $R_{\delta i}$, that represents partial route δ ending at node i with a vector containing $|N_\delta|$ binary values, where $|N_\delta|$ is the size of all nodes (excluding starting and ending ones). During the extension of label $\Lambda_{\delta i}$ to node j, the j^{th} element of this vector is set to 1. In case node j has already been visited in partial path δ (meaning that the j^{th} element of $R_{\delta i}$ is equal to 1), then label $\Lambda_{\delta i}$ is not extended to node j and, thus, the new label is not created. This process leads to the generation of elementary paths, i.e. it avoids re-visiting the same nodes.

Dominance Criteria

In order to avoid enumerating all feasible paths, dominance rules are applied to eliminate (discard) labels that are not Pareto-optimal and, therefore, cannot yield an optimal path. Eliminating labels improves significantly the computational efficiency of the solution approach. To do so, we have employed applicable dominance criteria from the literature for ESPPTWCC. We have also proposed additional dominance criteria that are particular to the problem in hand.

Given two labels, $\Lambda_{\delta' i}$ and $\Lambda_{\delta'' i}$ representing two different partial paths δ' and δ'' ending at the same vertex i, $\Lambda_{\delta' i}$ dominates $\Lambda_{\delta'' i}$ (i.e. δ'' is disregarded) if $\Lambda_{\delta' i} \leq \Lambda_{\delta'' i}$ (component-wise) and the inequality is strict for at least one component. In particular, the following inequalities must hold for the label components:

$$\tilde{c}_{\delta' i} \leq \tilde{c}_{\delta'' i} \tag{4.27}$$

$$t_{\delta' j} \leq t_{\delta'' j} \tag{4.28}$$

$$d_{\delta' j} \leq d_{\delta'' j} \tag{4.29}$$

$$\bar{\bar{c}}_{\delta' i} \leq \bar{\bar{c}}_{\delta'' i} \tag{4.30}$$

$$R_{\delta' i} \subseteq R_{\delta'' i} \tag{4.31}$$

Note that component $\bar{\bar{c}}_{\delta i}$ ensures optimality by including all C orders in a path, when needed. Note that this additional dominance criterion does not violate optimality when the associated ESPPTWCC is solved within a full column generation scheme, since it eliminates labels that lead to routes with higher reduced cost. Additionally, for the case of Ψ_{K+1} sub-problem, the equilibrium cost will be always zero (0), since $C = \emptyset$ and, thus, there is no affect to the overall dominance criteria and, consequently, to optimality.

Acceleration Techniques

In order to speed up the solution process, we have employed appropriate acceleration techniques from the literature. The acceleration techniques used in this work have been based on the work of Athanasopoulos (2011) for the Multi-Period VRP. Table 4.1 below lists the acceleration techniques adopted, which are briefly discussed below.

Table 4.1. Acceleration techniques for the solution of ESPPTWCC for the DVRPMB

Acceleration Technique	Reference
Unreachable nodes	Feillet *et al.* (2004; 2005), Chabrier (2006)
Limited Discrepancy Search (LDS)	Feillet *et al.* (2005)
Buckets / Storing Processed Labels	Larsen (2001), Chabrier (2006)
Early Termination Criterion	Larsen (2001), Chabrier (2006)
Parallel Implementation	

Unreachable nodes

Unreachable vertices, as defined by Feillet *et al.* (2004), are "*vertices that cannot be reached anymore due to resource constraints or because they have already been visited*". On top of the typical implementation of the unreachable nodes process, which discards non-feasible labels, we employ this technique for the solution of the SP for vehicles *en route*. Recall that in the sub-problems related to this case, all committed orders have to be included in the final path; thus, we assume that if at least one committed order is denoted as unreachable, then this label is considered as infeasible and is discarded from the set of labels $B(i)$ to be extended, even at a very early stage of the label generation process.

Limited Discrepancy Search (LDS)

This is a tree search method, initially developed by Harvey and Ginsberg (1995) for Constraint Programming. Our implementation follows the LDS framework successfully incorporated by Feillet *et al.* (2005) for solving the ESPPTWCC. The algorithm works as follows:

For each vertex $i \in N$, the m "closest" customers (referred to as "*good neighbors*") are chosen and included in a set $\mathcal{H}(i)$; proximity is measured by the value of the reduced cost. Extending a label to a node h that is not included in $\mathcal{H}(i)$ imposes a penalty (γ_{ih}) equal to 1, otherwise the penalty equals zero. Thus, every partial path δ ending at vertex i, i.e. every label in $B(i)$, is assigned a penalty value. The ending node (depot) is always considered as a good neighbor; also, for the source node μ', $m = |N|$. At the beginning of the algorithm, the acceptable cumulative penalty (denoted as CP) for a partial path δ (corresponding to a label $\Lambda_{\delta i}$), is set to zero and thus, labels are extended only to good neighbors, i.e. $\sum_{(i,j)\in r} \gamma_{ij} = 0$. This means that only arcs with $\gamma_{ij} = 0$ are selected. If there are any negative reduced cost routes (columns) after expanding all labels, the ESPPTWCC terminates and passes the related routes to the RMP. If not, CP is increased by 1 and labels with $\gamma_{ij} = 1$ are also allowed. An upper bound CP_{max} is defined, beyond which CP may not be increased. If CP_{max} has been reached and no negative reduced cost columns have been generated, then the operation terminates.

Buckets / Storing Processed Labels

As already mentioned, in our implementation we use two different structures for storing labels ending at vertex i; i.e., $B(i)$ that contains all non-processed labels, and $\check{P}(i)$ that stores all processed labels (i.e. labels that have been extended to all successors). Storing the labels in $\check{P}(i)$ supports the solution process, since these labels can be considered in the dominance checks and may discard (eliminate) non-processed labels not yet extended to all successors. A label $\Lambda_{\delta i}$ is checked if it dominates, or is dominated by, other labels within $B(i)$ and $\check{P}(i)$. A new label can eliminate labels from both sets $B(i)$ and $\check{P}(i)$ or may be eliminated by the labels in these sets. When a non-processed label $\Lambda_{\delta i}$ is eliminated by a label within $\check{P}(i)$, then it is not extended further. This process enhances the efficiency of the algorithm since more labels can be discarded during the dominance checks.

Early Termination Criterion

Many researchers terminate the solution process of the sub-problem when a predefined number of negative cost columns (routes) have been reached. Although this technique does not guarantee optimality for the sub-problem, the optimality of the global algorithm is still maintained, due to the iterations of the global algorithm between the RMP and the SPs. In our case, we terminate the solution process when at least 300 feasible columns (routes) with negative reduced cost have been found at each of the sub-problems $\Psi_k, \forall k = 1,2, \dots, |K| + 1$.

Parallel Implementation

Since the $|K| + 1$ problems of the ESPPTWCC are independent, they can be solved in parallel, resulting to gains in computational efficiency (see Fig. 4.2).

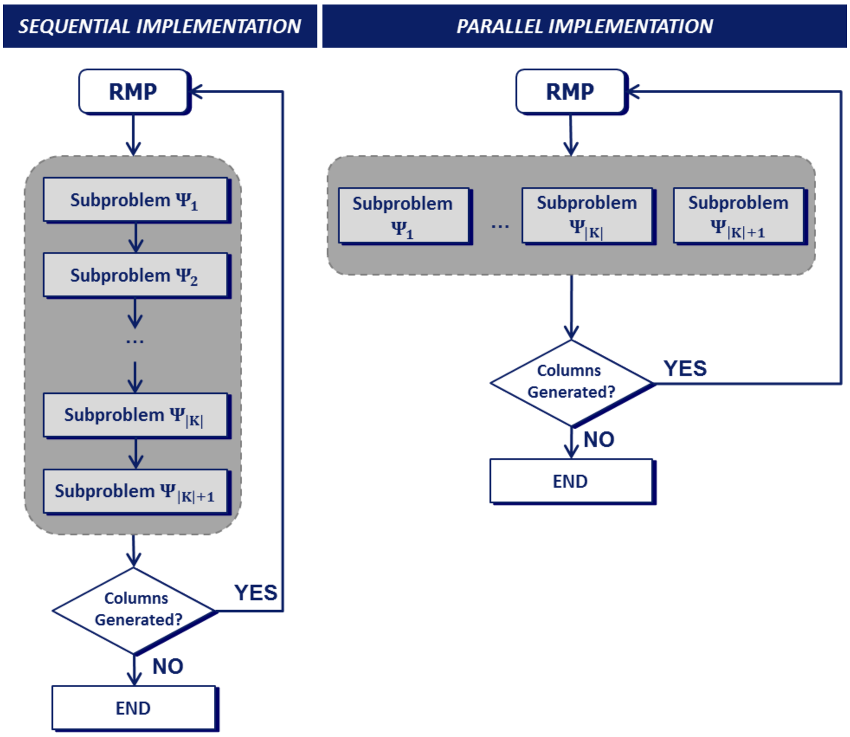


Figure 4.2. Sequential and parallel implementations of the Pricing Sub-problem

4.5. The integrated column generation scheme

Figure 4.3 shows the complete column generation framework, which integrates the RMP with the $|K| + 1$ sub-problems. The solution of the RMP provides the associated shadow prices, in addition to the cost and relevant routes. The former are provided to the $|K| + 1$ sub-problems, and are used to compute the modified costs $c'_{ij} = c_{ij} - \pi_i$ for each $(i, j) \in A$. These modified costs are the elements of the modified cost matrix in ESPPTWCC.

On the other hand, solving each sub-problem generates a set of negative cost routes. These routes (suitably represented as columns) are provided to the RMP and added to the existing routes/columns of the problem. The solution process terminates when no routes with negative cost can be generated by any sub-problem, matching the classical termination procedure of the simplex method. The minimum cost solution from the last RMP is the optimal solution.

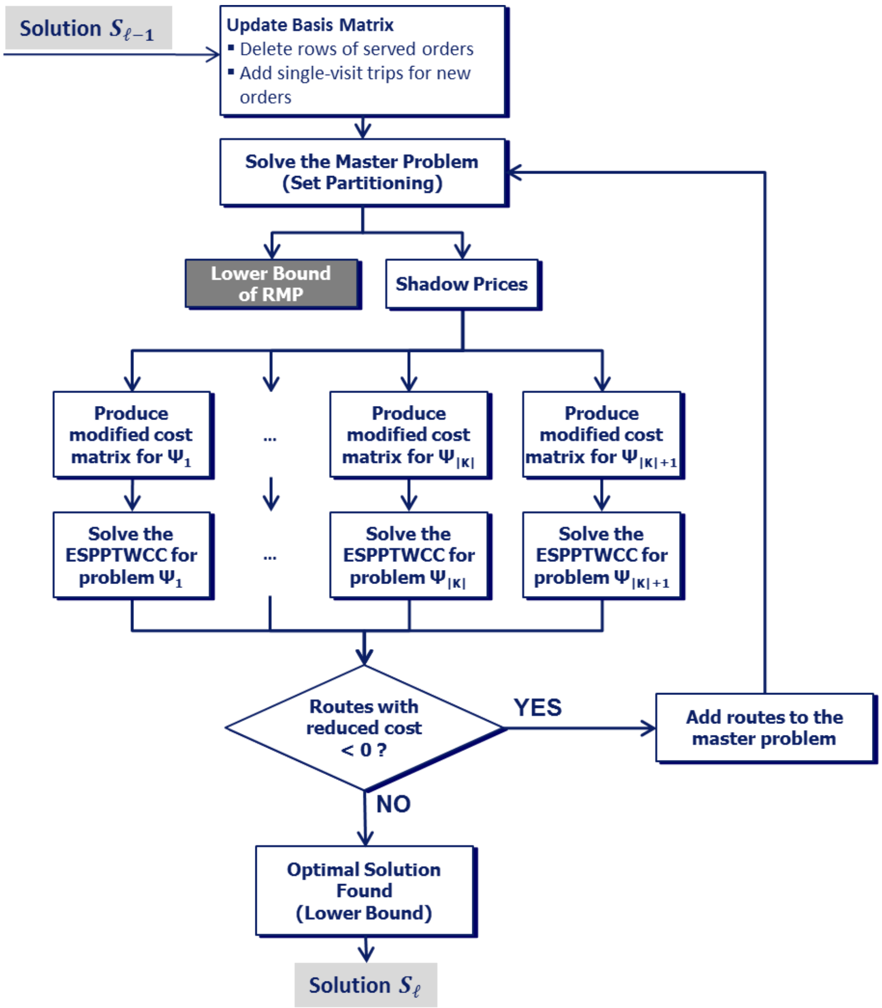


Figure 4.3. The column generation procedure at each re-optimization cycle

Example: single iteration of the column generation procedure

Consider an example of two vehicles ($K = \{1,2\}$), with four (4) offline (delivery) orders assigned to each vehicle. During re-optimization timestamp T_1, vehicle K_1 is located at customer 2, vehicle K_2 is located at customer 6 and three (3) new DO ($\{a, b, c\}$) have arrived in the interval $[0, T_1]$, as shown in Figure 4.4.

The first step in solving the related re-optimization problem is to create the initial basis, i.e. the Restricted Master Problem (RMP). To do so, we construct a matrix comprising the route columns of the initial plan; naturally all rows that correspond to the customers already served are deleted from this matrix. New columns corresponding to single-visit routes for the three DO are also included. This forms the first RMP which comprises of five (5) columns, $R_i, i = 1, \dots, 5$ each associated with a cost c_i ($i = 1, \dots, 5$). The RMP is solved using the Revised Simplex Method, and the dual prices produced for each customer generate the modified cost matrix for each sub-problem.

Thus, 2 + 1 sub-problems are solved. Each of the first two sub-problems (Ψ_1, Ψ_2) consider the corresponding set of committed orders $C_k, k = 1,2$ plus the set of all available DO (flexible orders), i.e. $N_k = C_k \cup F, k = 1,2$ (for example, $N_1 = \{3,4, a, b, c\}$). The third sub-problem considers only the available DO, i.e. $N_3 = \{a, b, c\}$. Sub-problems Ψ_1, Ψ_2 will provide columns that assign flexible orders to vehicles *en route*, while sub-problem Ψ_3 will provide columns that assign flexible orders to vehicles located at the depot.

Fig. 4.4 illustrates indicative columns generated by sub-problems Ψ_1, Ψ_2; based on the Figure, these SPs generated columns such that: a) each vehicle *en route* is able to serve all flexible orders (columns $R_1^{'}$ and $R_2^{'}$), and b) each vehicle is able to serve only a portion of flexible orders (columns $R_1^{''}$ and $R_2^{''}$). On the other hand, sub-problem Ψ_3 generates only one column that assigns all flexible orders to a single vehicle to be deployed from the depot (column $R_3^{'''}$). These columns are then added to the RMP which is solved again. This procedure is performed iteratively until no more negative cost columns are found by any sub-problem.

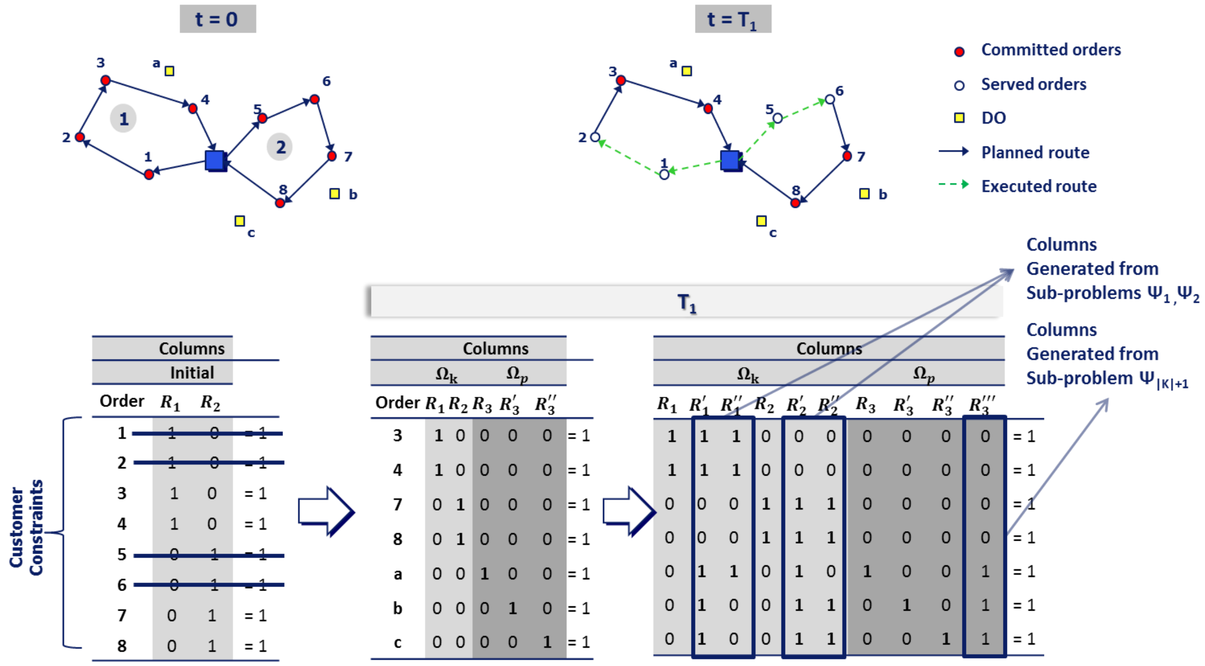


Figure 4.4. Example of the column generation process

4.6. Solving the integer problem (Branch-and-Price)

In case the solution of the column generation procedure is fractional, it provides a lower bound to the integer optimal solution. To obtain the latter, the column generation algorithm is embedded into a branch-and-bound ($B\&B$) search scheme, which is implemented using a *best-*

first strategy. In this case, column generation is used to compute lower bounds at each node of the branch-and-bound search tree. (Branch and Price – B&P).

The B&P procedure is initialized by obtaining the overall Lower Bound (LB) (as described previously) and a Global Upper Bound (GUB), which usually refers to the best known integer solution. GUB is originally set equal to a very large number (M). If LB does not correspond to an integer solution, the *Branching Policy* is triggered, which, given a fractional solution, divides the feasible solution space into two subspaces. Each subspace can be seen as a new node in the $B\&P$ tree and is further explored separately. Explored $B\&P$ nodes are discarded, while new nodes are added in the list of unprocessed nodes. GUB is updated based on improved integer solutions, if such solutions are found. Following the procedure, the next node to be explored from the unexplored node pool is selected by the *Node Selection* policy. The procedure terminates when all nodes from the pool have been explored, or when the LB of all those nodes is larger than the GUB.

Branching Policy

As proposed by several authors, we branch on arc flow variables $\psi_{ij}, (i,j) \in A$, i.e.:

$$\psi_{ij} = \sum_{r \in \Omega'} p_{ijr} y_r \tag{4.32}$$

where p_{ijr} denotes a binary variable equal to 1 if and only if route r traverses arc (i,j). The first subspace of the B&P scheme (i.e. $\psi_{ij} = 0$) is defined by an additional constraint which does not allow arc (i,j) to participate in the solution (i.e. remove from the master problem all variables y_r if route r contains arc (i,j)). The second subspace ($\psi_{ij} = 1$) forces arc (i,j) to be part of the solution. A major advantage of this strategy is that it can be easily implemented without adding new constraints to the Master Problem.

Considering the first subspace, all routes containing arc (i,j) are discarded, and coefficients c_{ij} are set to ∞. In the second subspace, all routes containing customers i and j that are not in sequence are discarded from the current RMP of the father node, and all cost coefficients c_{ih} and c_{hj}, $\forall h \neq \{i,j\}$ used in the sub-problem, are set to ∞. These two modifications will allow direct connections only from customer i to customer j.

Variable Selection Policy

To select the variable to branch on, we first identify the arc with the most fractional flow value (i.e. value ϱ, for which $\lceil \varrho \rceil - \varrho$ is closest to 0.5). Every arc $(i,j)_r$ of the routes r that participate in solution X, takes the corresponding value of variable x_r; the values of the same arcs are summed up and the arc with the most fractional part is then selected to be branched.

Node Selection Policy

Many existing policies are available in the literature, such as depth-first, best-first, width-first and depth first with backtracking (see Larsen, 2001; Lee and Mitchell, 2001). We have employed the Best-First approach, which is most commonly used in the literature. This policy selects to explore the node with the minimum LB among all nodes of the tree.

4.7. A heuristic-based column generation approach

Given the requirements for time efficiency of the solution process, especially for practical cases with extended solution space (e.g. without time windows), we propose a heuristic procedure to generate negative cost columns to enter the RMP instead of solving the ESPPRC to optimality (which is NP-hard). Thus, we propose a heuristic to solve the pricing sub-problem described in Section 4.4. The rest of the branch-and-price framework described in previous Sections remains intact.

The proposed CG-based heuristic requires distinct approaches for sub-problems $\Psi_k, k \in K$ and sub-problem $\Psi_{|K|+1}$. For the former sub-problems, an efficient (but not optimal) approach may result by finding the minimal cost of inserting each one of the flexible orders to each one of the available routes. The $\Psi_{|K|+1}$ sub-problem may be dealt as a new independent vehicle routing problem. Both are further explained below.

4.7.1. Generating columns for vehicles *en route*

To generate new columns with negative reduced costs, we use a local search procedure to modify the columns of the initial basis (Ω_k columns). The reason we utilize such columns is that each trip in the basis has zero reduced cost, and if such a trip is modified appropriately, it is likely to generate new trips with negative reduced cost. The modification is performed using a *cheapest insertion algorithm*, which tries to incorporate in a least-cost fashion each DO to each candidate column. For this insertion we use the flexible order-column combination that results in the minimum reduced cost; the latter is the difference between the reduced costs prior and after the order insertion. Let c_s be the cost (distance) of a column $s(k), k \in K$ prior to the insertion of flexible order f, and let c_{s_f} be the post-insertion cost. Also let RC_s and RC_{s_f} the respective reduced costs. The insertion criterion is provided by the following Equation:

$$G(s,f) = RC_{s_f} - RC_s = \left(c_{s_f} - \sum\nolimits_{a\in S_f} \pi_a\right) - \left(c_s - \sum\nolimits_{a\in S} \pi_a\right) \\ = \left(c_{s_f} - \sum\nolimits_{a\in S} \pi_a - \pi_f\right) - \left(c_s - \sum\nolimits_{a\in S} \pi_a\right) = c_{s_f} - c_s - \pi_f \tag{4.33}$$

where $S = \{s(1), s(2), \dots, s(K)\}$ denotes the set of columns in the optimal basis corresponding to columns for vehicles *en route* at re-optimization cycle T_ℓ, and π_α, π_f refer to the dual prices of each order $a \in S$ and $f \in F$, respectively. Using this criterion, each order in F is tested for insertion in all possible positions of each column of the initial basis. Columns with negative reduced cost that are generated during the iterations of this process are maintained as candidates in a pool of columns to be added to the RMP. This operation is terminated when no negative cost columns can be found, or when all orders in F have been tested for insertion. The operation is described in detail below; the pseudocode of the algorithm is given in Figure 4.5.

Preliminary Stage: Initialization

Let the set of new columns to be added to the RMP (column pool) be $\Omega'' = \emptyset$. Also let G and RC denote matrices of size $|K| \times |F|$. The elements of matrices G and RC store the information related to the values of the insertion criterion of Eq. (4.33) and the total reduced costs, respectively, for each assignment of flexible order $f \in F$ to each one of the vehicle routes $k \in K$. At the initial state, all elements of the G and RC matrices are set to a large positive number.

Stage 1: Generating New Columns

Step 1.1. Select a column $s(k) \in S$. This column corresponds to a route that starts from the current position of the vehicle μ_k, serves orders $O(s(k))$ and ends at the depot. Thus, there are $|O(s(k))| + 1$ possible positions, denoted as $\tilde{A}(s_k)$, where F orders can be inserted. Let the cost of this trip be $c_{s(k)}$.

*Step 1.2.*For each flexible order $f \in F$, set the current best cost $\bar{c}_f^{s(k)} = Z$ (where Z a very large positive number) and perform the steps below. If there is no such f, i.e. $F = \emptyset$, go to Stage 2.

1.2a. Try to incorporate f on a possible position $\nu \in \widetilde{A}(s_k)$. If $\tilde{A}(s_k) = \emptyset$, go to Step 1.2. If path $s_k^{f\nu}$ that incorporates flexible order f in position ν of column s_k satisfies all feasibility constraints (i.e. time windows and capacity), go to Step 1.2b, otherwise repeat Step 1.2a until all positions $\nu \in \widetilde{A}(s_k)$ are examined.

1.2b. Improve trip $s_k^{f\nu}$ with a 2-opt post-optimization procedure (Li, 1965). Let $c_k^{f\nu}$ be the cost of this improved trip.

1.2c. If the reduced cost $r_k^{f_v}$ of trip $s_k^{f_v}$ (i.e., $r_k^{f_v} = c_k^{f_v} - \sum_{\alpha \in s(k)} \pi_\alpha$) is negative, then add trip $s_k^{f_v}$ in the column pool, i.e. $\Omega'' = \Omega'' \cup \{s_k^{f_v}\}$

1.2d. If the current inclusion of flexible order f does not provide a better cost than the previous ones, i.e., $c_k^{f_v} \geq \bar{c}_f^{s(k)}$, return to Step 1.2a. Otherwise, set element (s, f) of matrix RC equal to the reduced cost of trip $s_k^{f_v}$ and the same element of matrix G equal to $(c_k^{f_v} - c_{s(k)}) - \pi_f$. Finally, set the current best cost to be $\bar{c}_f^{s(k)} = c_k^{f_v}$.

Stage 2: Pseudo-assignment of the "cheapest" pick-up order

Step 2.1. If all elements of matrix RC are non-negative (i.e. there is no insertion operation that yields a negative reduced cost from Stage 1), terminate the procedure and return the column pool Ω'' generated during the process. Otherwise, go to Step 2.2.

Step 2.2. Select order $f^* \in F$ to be pseudo-assigned in the current plan such that $g_{s_p} = \min(G(s,f) \mid \forall s \in S, \forall f \in F)$; i.e., the order that corresponds to the minimum element of matrix G. Denote as $s(k^*)$ the column that satisfies the above statement.

Step 2.3. Update all problem data according to the pseudo-assignment of Step 2.2, i.e. the set of orders $O(s(k^*)) = O\big(s(k^*)\big) \cup \{f^*\}$ and the set of flexible orders $F = F \backslash \{f^*\}$. Finally, update matrices G and RC by deleting the column that corresponds to f^*.

Figure 4.5 below provides a pseudo-code of the above heuristic procedure for the generation of new columns to be added to the RMP for vehicles *en route*.

Algorithm 1: Heuristic for generating columns for vehicles *en route*

$\Omega'' = \emptyset$; // New set of columns generated by the procedure
$G = \emptyset$; // Matrix that stores insertion costs of order f to each column s
$RC = \emptyset$ // Matrix that stores the reduced cost of order f to each column s
While $F \neq \emptyset$
  For each column $s(k) \in S, k \in K$ **do**
    $c_{s(k)}$ → Cost of column $s(k)$
    For each order $f \in F$ **do**
      $\bar{c}_f^{s(k)} = Z$ // Best cost of including order f in column $s(k)$
      For *every feasible arc* ν *in path* s *for inserting order* $f \in F$ ***do***
        Apply insertion of order f *in path* s
        [path(ν)] = Apply 2-opt improvement on this temporary path
        $s_k^{f\nu}$ *= path(ν)* // *column* s *with order* f *on arc* ν *after 2-opt*
        $c_k^{f\nu}$ *= Cost of path(ν)*
        Compute reduced cost $r_k^{f\nu} = c_k^{f\nu} - \sum_{\alpha \in s(\kappa)} \pi_\alpha$
        If $r_k^{f\nu} < 0$ ***then***
          $\Omega'' = \Omega'' \cup \{s_f^{\nu}\}$
        End
        If $c_k^{f\nu} < \bar{c}_f^{s(k)}$ ***then***
          $G(s,f) = (c_k^{f\nu} - c_{s(k)}) - \pi_f$
          $RC(s,f) = r_{s_f^{\nu}}$
          $\bar{c}_f^{s(k)} = c_k^{f\nu}$
        End
      End
    End
  End
  If $all(RC) \geq 0$
    terminate procedure and return $\boldsymbol{\Omega''}$
  Else
    Find $s(k')$ and $f^* \in N$ such that $g_{s_{p'}} = \min(G\{s\}\{f\} \mid \forall s \in S, \forall f \in F)$
    Update column $s(k^*) = s(k^*) \cup \{f^*\}$ // in the best feasible place
    Update matrices G and RC
    Update set of F orders → $F = F \setminus \{f^*\}$
  End
End

Figure 4.5. Pseudo-code of heuristic approach for generating columns for vehicles *en route*

Implementation techniques for computational efficiency

Since the procedure seeks to insert a flexible order at each possible position of a column, followed by a post-optimization procedure, it is probable that it may generate multiple identical columns. For example, the assignment of flexible order f in the $\nu - th$ position of the route that is represented by a column s can also be the result of the 2-opt procedure when order φ is tested for insertion at another position of the same route. In order to avoid generating identical columns and provide RMP with unnecessary information, we tag each generated path with an appropriate value, denoted as *representative information*, in order to discard them if they've been encountered again. This is carried out by employing *Hashing Functions*, introduced by Juliff (1990) and successfully implemented to a tabu search metaheuristic for a VRP variant by Osman and Wassan (2002). Hashing functions require a unique code to be computed for each solution. In our case, code H_s is calculated for each generated column as the product of the column index with the sum of the products of the customer index times the total number of orders of the related path, i.e.

$$H_s = s \times \sum_{u_j \in O_s} u_j \times |O_s| \tag{4.34}$$

where O_s is the set of orders in column s, $|O_s|$ represents the total number of orders in column s, and u_j represents the customer index of that column. The above equation ensures different codes or records for almost all generated columns with different characteristics; any column that is found with the same code is discarded.

A second simple and efficient acceleration technique deals with those flexible orders that can never be included in a certain column. This technique is almost similar to the *unreachable nodes* described in Section 4.4. Before the initialization of the procedure, each flexible order is checked for feasibility of inclusion in each one of the available columns. Once an order fails w.r.t. at least one of the feasibility criteria, i.e. time windows, capacity or total permitted length of the trip, it is no longer considered a candidate for the corresponding column.

4.7.2. Generating columns for vehicles located at the depot

The solution of sub-problem $\Psi_{|K|+1}$ for generating columns for vehicles located at the depot is possible within the framework described in Section 4.4, since the resources (remaining time horizon, capacity, etc.) are relatively limited at each re-optimization cycle. This, of course, holds when the number of orders in set F is relatively limited. Thus, if the number of F orders is less than or equal to a reasonably small number, e.g. $|F| \le \Xi$, we use the label correcting algorithm as described in Section 4.4. For $|F| > \Xi$, we apply the same algorithm but we exclude path elementarity from the dominance criteria; i.e. a label can be eliminated by another label even if the dominator is not a subtour of the dominated one. This may speed up the solution process, since it eliminates a significant number of columns, but cannot ensure that all feasible Ω_p will be generated, and the optimum will be reached.

Chapter 5: RE-OPTIMIZATION STRATEGIES FOR THE CASE OF UNLIMITED VEHICLE FLEET

One of the most critical elements in a re-optimization process is the timing of re-optimization, i.e. the time(s) at which the current plan is recomputed in order to incorporate the up-to-date information. In this Chapter, we investigate the *re-optimization strategy*, that is the combination of i) the length of the re-optimization cycle (i.e. when to re-plan, hereafter the *re-optimization policy*) and, ii) the part of the plan that is released to the drivers for implementation (hereafter the *implementation tactic*). The main purpose is to analyze alternative strategies and propose guidelines under various operational settings with respect to key problem characteristics (time windows, degree of dynamism, time of occurrence of dynamic orders, etc.).

Significant issues related to re-optimization in dynamic routing are overviewed in Section 5.1. In Section 5.2 we propose and analyze several re-optimization strategies. Theoretical insights regarding these strategies are discussed in Section 5.3. Section 5.4 presents the experimental investigation of the proposed strategies under various operational characteristics. Finally, Section 5.5 summarizes the key findings of this part of the study.

5.1 Re-optimization and the routing environment

The selection of the appropriate re-optimization strategy may be affected by the characteristics of the routing environment. In this Section we discuss significant aspects regarding key factors and how they affect the routing results of the overall dynamic problem.

The **re-optimization frequency** (length of the re-optimization cycle) in dynamic routing should strike an appropriate balance. Very frequent re-optimization (short re-optimization cycles) may limit the solution quality of the overall (long-term) problem, since it may not take advantage of combinations of newly arrived requests. On the other hand, infrequent re-optimization (long re-optimization cycles) may limit the dispatcher's options since a larger portion of the route has been completed during previous re-optimization cycles, and fewer options are available for incorporating the newly arrived requests.

To illustrate the significance of the re-optimization frequency, we provide in Figures 5.1 and 5.2 two examples, each reflecting potential inefficiencies when adopting short and longer re-optimization cycles, respectively. For the former example (Fig. 5.1), two vehicles *en route* are scheduled to serve committed orders. Two re-optimization policies are tested. In the first policy, shown in Fig. 5.1a, re-optimization is applied whenever new information is received (every time a new order is received by the dispatchers). The Figure presents the evolution of the system in two successive re-optimization cycles. In the second policy, re-optimization is applied after two DO have been received (Fig. 5.2b). The Figure presents the evolution of the system after a single cycle. In this example, the second policy yields better overall results due to the opportunity provided to the algorithm to consider the allocation of both DO at the same time, and combine them appropriately.

The second example considers a single vehicle executing a planned route (see Fig. 5.2) and illustrates the reverse case, in which re-optimizing after each DO is received yields better results than re-optimizing every two DO. Figure 5.2a is related to the policy in which re-optimization is performed upon the receipt of each DO (two re-optimization cycles), while Fig. 5.2b presents the result of the 2 DO re-optimization policy. In this case, the former policy yields superior results, since the portion of the route that favors the inclusion of DO A has not yet been completed under this policy.

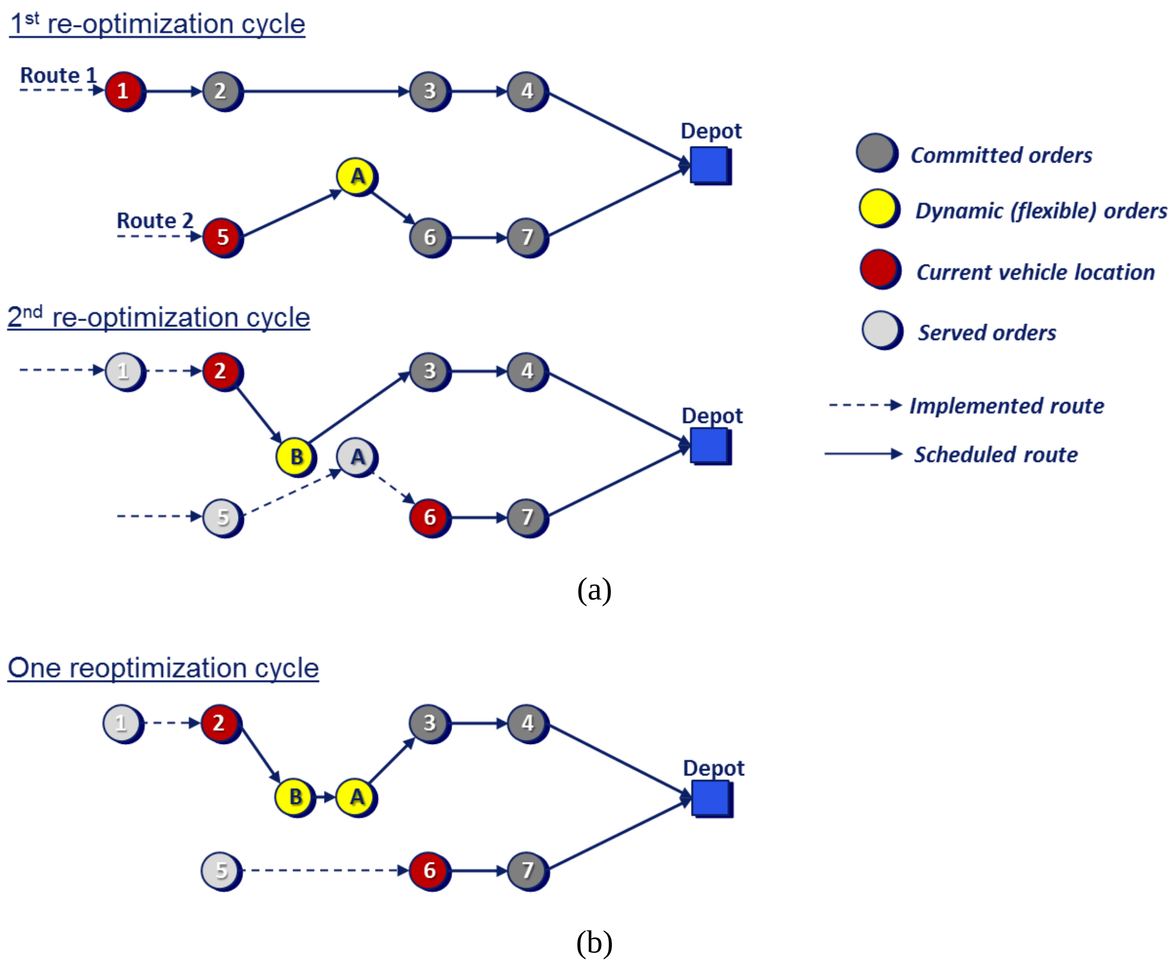
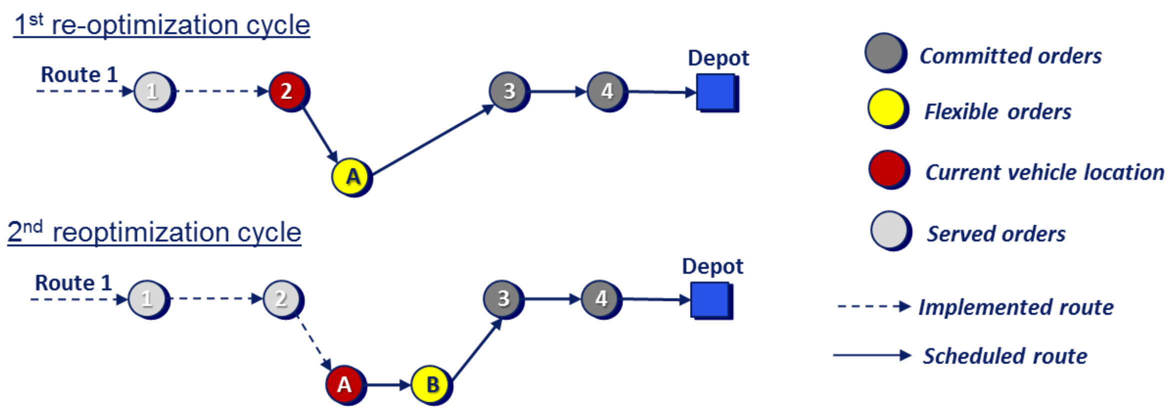
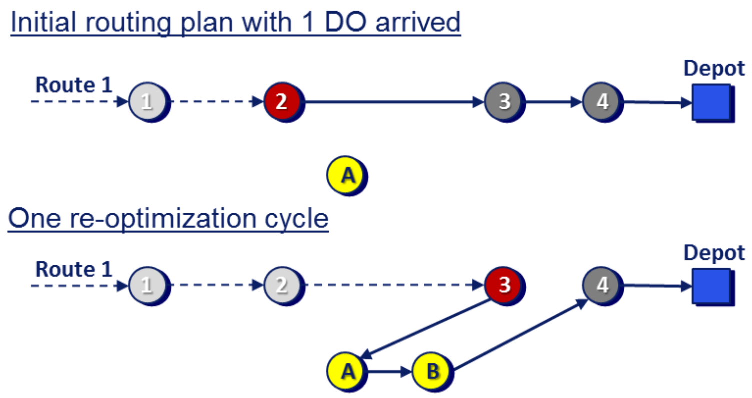


Figure 5.1. Example in which (a) re-optimizing upon the receipt of each DO yields an inferior result than (b) re-optimizing when both DO are received

The **implementation tactic** defines which DO are released to the fleet for implementation after the execution of re-optimization. It seems that releasing a DO only when it is absolutely necessary will provide more possibilities for DO combinations. However, as discussed in Chapter 3, there are some practical cases in which this may not be applicable and DO have to be released to the fleet immediately after re-optimization (and considered as committed). Consequently, in Section 5.2 we examine those two scenarios met in practice and we investigate their interaction with different re-optimization policies (frequency).



(a)



(b)

Figure 5.2. Example in which (a) re-optimizing upon the receipt of each DO yields a superior result than (b) re-optimizing when both DO are received

As mentioned above, it is anticipated that the selection of a re-optimization strategy may be significantly affected by the characteristics of the underlying routing environment. In particular, we focus on the following characteristics:

- *Geographical distribution of customers*: The spatial distribution of customer locations is essential in any type of vehicle routing system. For example, in the clustered case in which customers form distinct groups, the excess cost of an additional visit within the same cluster tends to be lower. Consequently, it is expected that in these cases, a larger portion of the route will be completed (more customers served) if infrequent re-optimization will be chosen. This may limit the allocation options of DO during future cycles.
- *Customer time windows (TW)*: The characteristics of TW is expected to significantly affect the solution quality with respect to the selection of the re-optimization strategy. For example, tight TW cases may limit the impact of the re-optimization strategies, since the solution space is significantly limited. On the other hand, wide TW provide more allocation options of the newly received DO in the current routing plan; this may favor the solution

under wider re-optimization intervals especially for cases where DO should be released immediately after re-optimization.

- *Degree of dynamism*: The number of DO with respect to the total number of orders may also affect the selection of re-optimization strategy. Higher degrees of dynamism may require higher re-optimization frequencies.

In this Chapter, we investigate the impact of the aforementioned characteristics on the selection of the re-optimization strategy. To do so, we apply the re-optimization approach of Chapter 4 under various re-optimization strategies and operating scenarios. The results of the analysis lead to guidelines regarding the appropriate re-optimization strategy with respect to the characteristics of the routing environment.

5.2 Re-optimization strategies

As mentioned above, the re-optimization strategy is defined by the re-optimization policy (frequency of re-optimization) and the implementation tactic (order release tactic).

We explore various re-optimization policies depending on the number of DO that have arrived between two successive re-optimization instances; that is:

- *Single-Request Re-optimization (SRR)*: Re-optimize upon the arrival of each DO
- *Number of Requests Re-optimization (NRR)*: Re-optimize after the arrival of a predefined number (more than 1) of DO (e.g. after three DO have been received)
- *Fixed-Time Re-optimization (FTR)*: Re-optimize at predefined time intervals (e.g. once per hour).

The aforementioned re-optimization policies are illustrated in Figure 5.3 (Larsen, 2000).

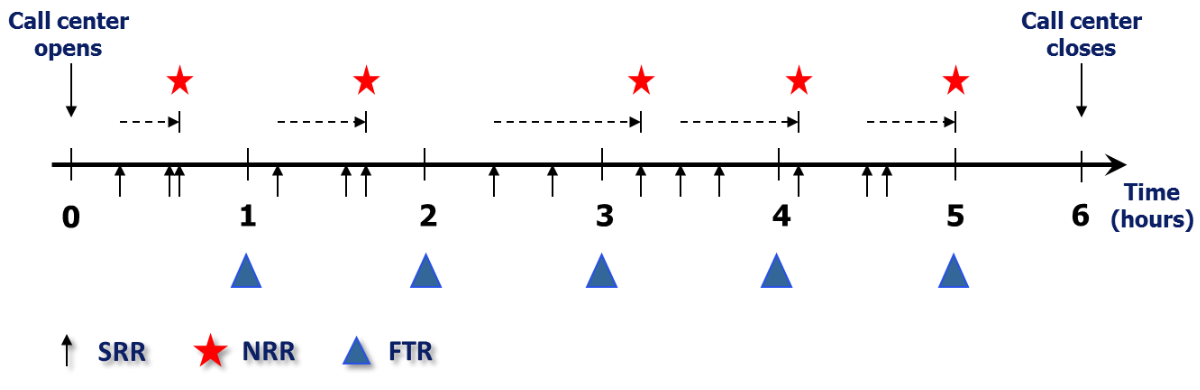


Figure 5.3. Illustration of re-optimization policies (NRR is applied for N=3; FTR is applied every hour)

In addition, we also explore two (2) *tactics* to implement the new plan:

- ❒ ***Full-Release tactic (FR)***: All re-optimized DO are released to the fleet immediately for implementation and they cannot be reassigned at later re-optimization cycles (see the discussion of Chapter 3, Section 3.2.1 on relevant practical cases, in which this tactic is applicable)
- ❒ ***Partial-Release tactic (PR)***: Only the DO scheduled for implementation prior to the next re-optimization cycle are released and the remaining DO are re-considered in the next cycle. In practical terms this means that not yet served DO up to the re-optimization timestamp are included in the F set as flexible orders.

Specifically, the FR tactic considers as flexible orders only DO that arrived during the interval $[T_{\ell-1}, T_\ell]$, while the PR tactic considers also DO arrived in $[T_0, T_{\ell-1}]$ but not served yet (see relevant discussion in Chapter 3.2). During the implementation under the FR tactic, the entire plan is released for implementation and has to be executed as designed (solution S_ℓ). For the PR tactic, only the DO allocated for the interval $[T_\ell, T_{\ell+1}]$ are released for implementation. The implementation of this tactic depends on the technology used; typically, the driver receives only the DO to be served next. The above considerations and differences between the two (2) tactics are illustrated in Figure 5.4.

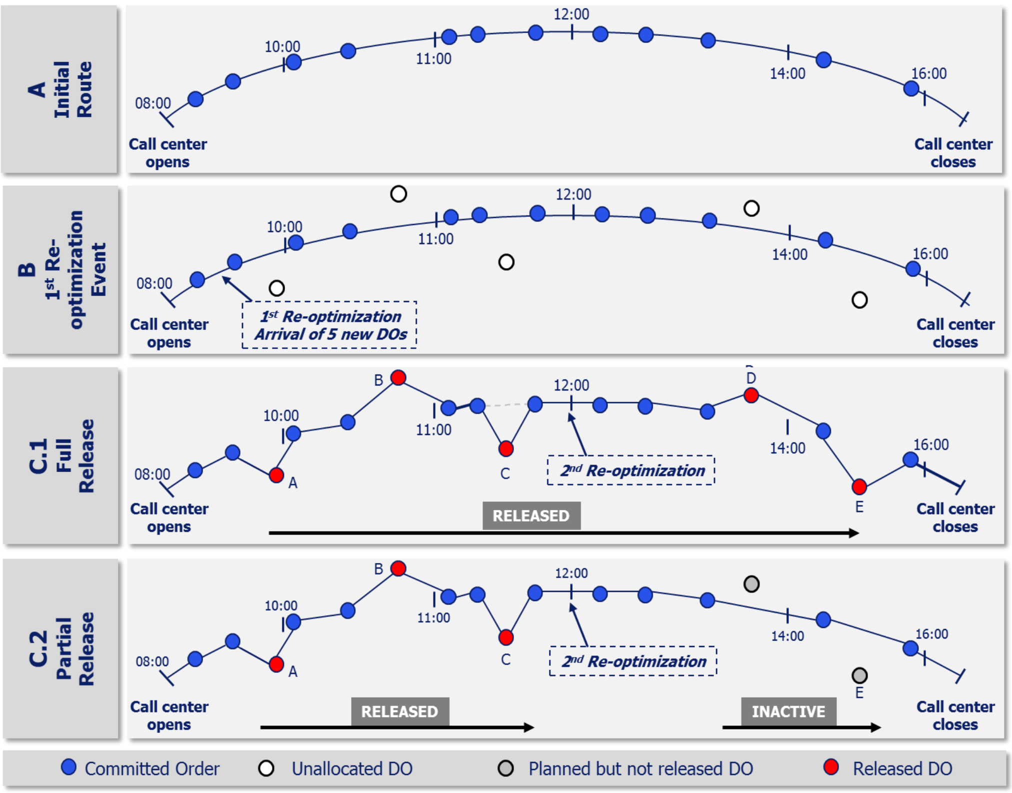


Figure 5.4. Full release vs. partial release tactic for a single re-optimization cycle and a single route

5.3 Theoretical insights for re-optimization strategies

It is reasonable to expect that the PR tactic is superior to FR. Below we examine in which cases this holds assuming the following conditions:

- All orders are served in the final solution
- Vehicles located at the depot are eligible to be dispatched at any $\ell > 0$
- Both release tactics are compared under the same number of re-optimization cycles
- An optimal method is used for re-optimization.

We should initially note that in the trivial case of a single-vehicle, both release tactics lead to identical results. This is due to the fact that although FR commits flexible orders for the next re-optimization cycles, the sequence of customer service within the route of each vehicle (the only one in this trivial case) is not committed; thus, the re-optimization state is the same for both tactics, which generate identical optimal solutions.

Claim 1: *It is guaranteed that the cost of the overall solution (for* $[T_0, T_{max}]$*) obtained under the PR tactic is always lower than or equal to the cost of the solution obtained under the FR tactic, for* $\ell < 3$.

Consider a simple example with $L = 2$ re-optimization cycles, K vehicles scheduled to be dispatched at time T_0, and K^d available vehicles at the depot eligible to be dispatched at any $\ell > 0$. Let K_ℓ denote the vehicles *en route* considered at each cycle ℓ (comprising of vehicles that have not completed their assignments). Let $RP(\omega, \ell)$ denote the re-optimization problem for each implementation tactic $\omega \in \{FR, PR\}$ with total routing cost $O(\omega, \ell)$. Note that $O(\omega, \ell) = O_p(\omega, \ell) + O_f(\omega, \ell)$, where $O_p(\omega, \ell)$ denotes the cost of the already completed portion of the routes up to T_ℓ, and $O_f(\omega, \ell)$ the cost of the solution for $[T_\ell, T_{max}]$.

The feasible space of each $RP(\omega, \ell)$, $\ell > 0$ may be formed by considering (a) all feasible combinations of assigning the flexible orders among the vehicles *en route* (Ψ_{K_ℓ} sub-problems) and the vehicles located at depot ($\Psi_{|K_\ell|+1}$ sub-problem), and (b) for each sub-problem, all feasible sequences of customer orders assigned to each vehicle.

The problem solution at each ℓ is affected by the re-optimization state, which is comprised of i) the set of committed orders $C_k(\omega, \ell), k = \{1,2, \dots, K_\ell, K_\ell + 1\}$, ii) the set of flexible orders $F(\omega, \ell)$, and iii) the current location of the vehicle(s). Note that there are no committed orders for the $\Psi_{|K_\ell|+1}$ sub-problem, i.e. $C_{K_\ell+1}(\omega, \ell) = \emptyset$.

During $\ell = 1$ both tactics consider the same re-optimization state and $O_p(FR, 1) = O_p(PR, 1)$; therefore, the related re-optimization problems (for PR and FR) are identical with identical solutions, and $O(FR, 1) = O(PR, 1)$.

During $\ell = 2$ (at time T_2), it holds that the current locations of the vehicles *en route* at T_2 are identical for both tactics and $O_p(FR, 2) = O_p(PR, 2)$. For each tactic, the related problems consider the sets of orders $N_k(\omega, 2) = C_k(\omega, 2) \cup F(\omega, 2), k = \{1,2, \dots, K_\ell, K_\ell + 1\}$; more explicitly:

- *FR-tactic*: $N_k(FR, 2) = C_k(FR, 2) \cup F(FR, 2) = [C_0^k(2) \cup F_k'(2)] \cup F_0(2)$, where $C_0^k(2)$ denotes the set of unserved static orders and $F_k'(2)$ the subset of DO arrived during $[T_0, T_1]$ and assigned to vehicle k but not yet served; $F_0(2)$, denotes new orders arrived during $[T_1, T_2]$. Note that $N_{K_\ell+1}(FR, 2) = F_0(2)$.
- *PR-tactic*: $N_k(PR, 2) = C_k(PR, 2) \cup F(PR, 2) = C_0^k(2) \cup [F'(2) \cup F_0(2)]$, where $F'(2)$ denotes all orders arrived during $[T_0, T_1]$ and not yet served. Also, $N_{K_\ell+1}(PR, 2) = F'(2) \cup F_0(2)$.

Since $F'(2) = \bigcup_{k \in K} F_k'(2)$, it is clear that $N_k(FR, 2) \subseteq N_k(PR, 2), \forall k \in \{1,2, \dots, K_\ell, K_\ell + 1\}$. Thus, the feasible subspace corresponding to the PR tactic is a superset of that of the FR tactic (only for $\ell = 2$), and $O_f(FR, 2) \leq O_f(PR, 2)$; consequently, $O(PR, 2) \leq O(PR, 2)$.

Based on the above, up to $\ell = 2$, the PR tactic will always provide superior or equivalent results.

For $\ell > 2$, however, such a comparison between the two tactics is not possible, since i) the state of the system at each re-optimization event is not the same and, ii) the cost $O_p(\omega, \ell)$ up to that event is, in general, different for each tactic.

Claim 2: *For $\ell \geq 3$, and if more than one vehicles are involved (dispatched at either $\ell = 0$ or at $\ell > 0$), it is not guaranteed that the overall routing cost under the PR tactic is lower or equal than the one obtained by the FR tactic.*

We will show this claim through a counter-example illustrated in Figure 5.5. At $\ell = 0$, two vehicles are planned to execute four (4) deliveries (customers 1, 2, 3, 4). During the course of implementing this plan, three (3) DO arrive and should be incorporated in the plan (customers 5, 6, 7). Re-optimization is triggered upon arrival of each DO. Table 5.1 provides the coordinates of all customers; the depot (denoted by node 0) is located at point (0,0).

Table 5.1. Customer coordinates for counter-example of Claim 2

Customer ID	Coordinates (X,Y)
1	$(5,2.5)$
2	$(10,5)$
3	$(10,-5)$
4	$(5,-2.5)$
5	$(10,0)$
6	$(10,10)$
7	$(5,0)$

Figure 5.5 illustrates two states per implementation tactic for $\ell > 0$; the state prior to re-optimization ("*Before*") and the state after re-optimization ("*After*").

At $\ell = 1$, both implementation tactics provide the same routing plans, as expected. At $\ell = 2$, order (6) has arrived. The $F_0(2)$ set for FR comprises only the new order 6, while for PR the $F_0(2)$ set includes all DO not yet served ($\{5,6\}$). PR generates a superior solution, since the flexible order set is a superset of the one considered by FR. However, for $\ell = 3$, the (initial) re-optimization states of the two implementation tactics are different. Thus, the two generated plans are different, and, in this case, the overall solution of the FR tactic is superior to that of the PR tactic. Table 5.2 presents the final routing costs after each re-optimization cycle for each implementation period ($[T_\ell, T_{max}]$) and the entire planning horizon ($[T_0, T_{max}]$).

Table 5.2. Routing costs under both tactics for three re-optimization cycles

	$\boldsymbol{FR - tactic}$		$\boldsymbol{PR - tactic}$	
	Cost $[\boldsymbol{T_\ell}, \boldsymbol{T_{max}}]$	**Cost** $[\boldsymbol{T_0}, \boldsymbol{T_{max}}]$	**Cost** $[\boldsymbol{T_\ell}, \boldsymbol{T_{max}}]$	**Cost** $[\boldsymbol{T_0}, \boldsymbol{T_{max}}]$
$\boldsymbol{\ell = 0}$	44.72	44.72	44.72	44.72
$\boldsymbol{\ell = 1}$	31.77	48.54	31.77	48.54
$\boldsymbol{\ell = 2}$	36.18	58.54	35.32	57.68
$\boldsymbol{\ell = 3}$	25.59	58.54	27.23	59.59

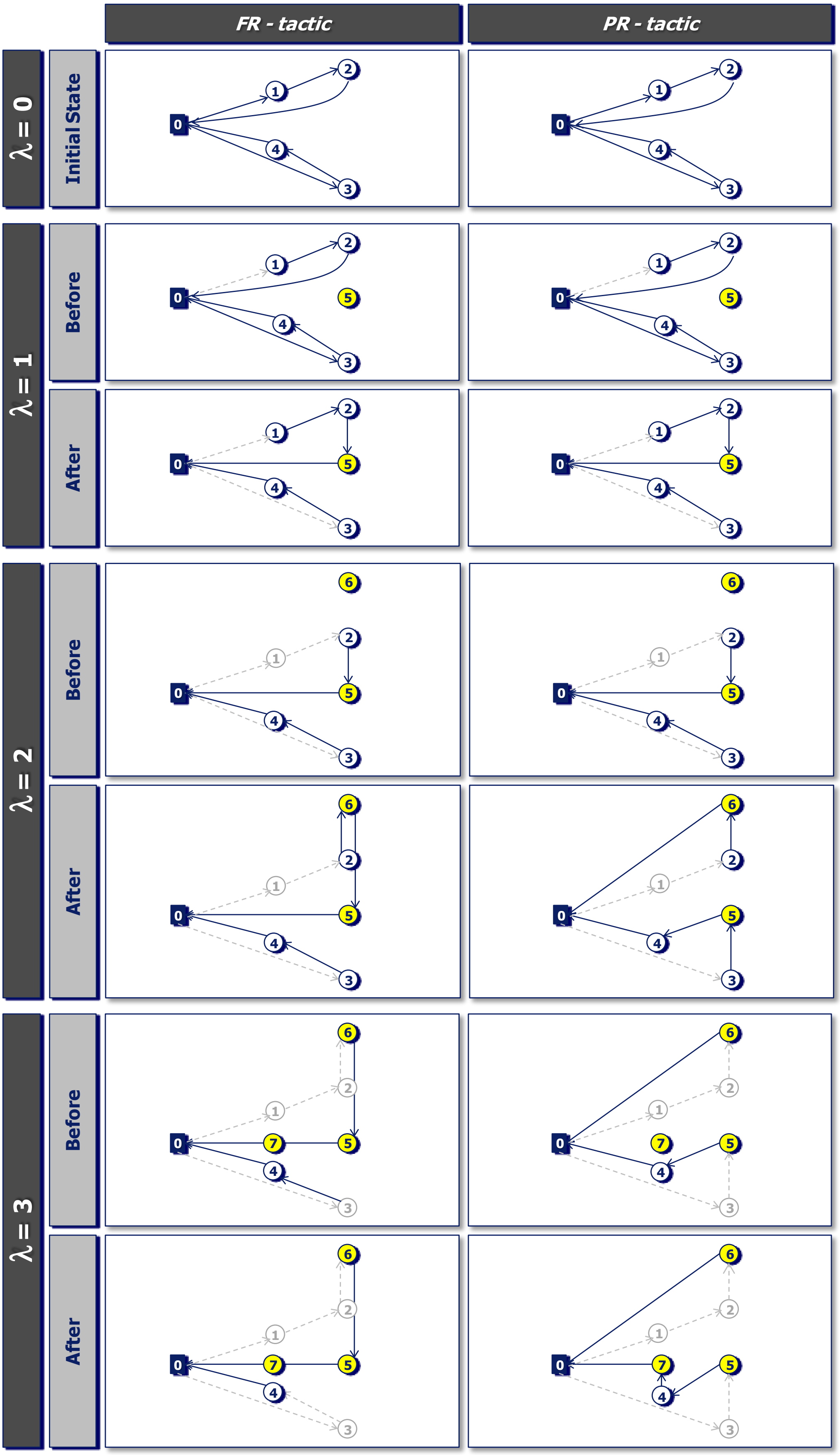


Figure 5.5. Planned and actual routes under both tactics for three re-optimization cycles (solid line is the planned route, dotted line the executed route)

5.4 Computational experiments

We study below the performance of the proposed re-optimization heuristic (Section 4.7) and of the proposed re-optimization strategies (policies and tactics) through extensive experimentation. In Section 5.4.1 we describe how the test problems were generated; in Section 5.4.2 we assess the performance of the proposed heuristic w.r.t. the exact B&P method. Finally, in Section 5.4.3 we investigate the performance of the re-optimization strategies. In this latter Section, in order to be able to report results under a unified solution framework, we employ the heuristic method described in Chapter 4.7 for all experiments. The experimental study was conducted using a Quad-Core Intel i7 processor of 2.8GHz and 4GB of RAM.

5.4.1 Experimental setup

5.4.1.1 The value of information

Measuring the solution efficiency of a dynamic optimization problem, such as the one addressed here, is not a straightforward task, as also discussed in Mitrovic-Minic *et al.* (2004) and Pillac *et al.* (2013). In this study we report the performance of the proposed method based on the so-called *value of information*, which was originally introduced by Mitrovic-Minic *et al.* (2004). Consider the DVRPMB instance $\mathcal{H}$ and the related static problem $\mathcal{H}_s$, in which all DO are known prior to vehicle dispatching (at $t = 0$). Then the value of information metric $V_{\mathcal{F}}$ corresponding to algorithm $\mathcal{F}$ when solving dynamic problem $\mathcal{H}$ is defined by the following expression:

$$V_{\mathcal{F}}(\mathcal{H}) = \frac{z_{\mathcal{F}}(\mathcal{H}) - z_{\mathcal{F}}(\mathcal{H}_s)}{z_{\mathcal{F}}(\mathcal{H}_s)} \times 100 \qquad (5.1)$$

where $z_{\mathcal{F}}(\mathcal{H})$ and $z_{\mathcal{F}}(\mathcal{H}_s)$ are the values of the objective function for dynamic problem $\mathcal{H}$ and for the related static problem $\mathcal{H}_s$, both solved by algorithm $\mathcal{F}$. Note that $\mathcal{F}$ is employed at each re-optimization step for $\mathcal{H}$, while $\mathcal{F}$ is employed once to solve $\mathcal{H}_s$.

5.4.1.2 Test instances

For the experimental study we have employed all R1, C1 and RC1 benchmark datasets of Solomon (1987). Furthermore, we employ datasets MR2, MC2 and MRC2 of Kontoravdis and Bard (1995), who have used Solomon's R2, C2 and RC2 datasets; the authors also reduced the original value of the vehicle capacity from 1000 to 250 units. Thus, effectively, our experimental investigation considers the full array of features of the Solomon benchmarks. In the latter, as shown in Fig. 5.6, the Cartesian coordinates of customers in the R configuration are randomly generated from a uniform distribution. Configuration C relates to clustered customers, whereas RC contains semi-clustered customers (i.e. a combination of clustered and randomly distributed customers). The datasets contain between 8 and 12 100-node instances; datasets R1, C1 and RC1 correspond to short-horizon problems requiring multiple vehicles with limited number of customers per route. In contrast, instances included in datasets MR2, MC2 and MRC2 consider long scheduling horizons and allow the assignment of many customers per vehicle.

In our experimental investigation, we have also employed instances vrpnc8 and vrpnc14 of Christofides *et al.* (1979) that have no TW for the uniform and clustered cases, respectively, but use the same customer coordinates of the R1 and C1 datasets. We designate these instances as R100 and C100, respectively.

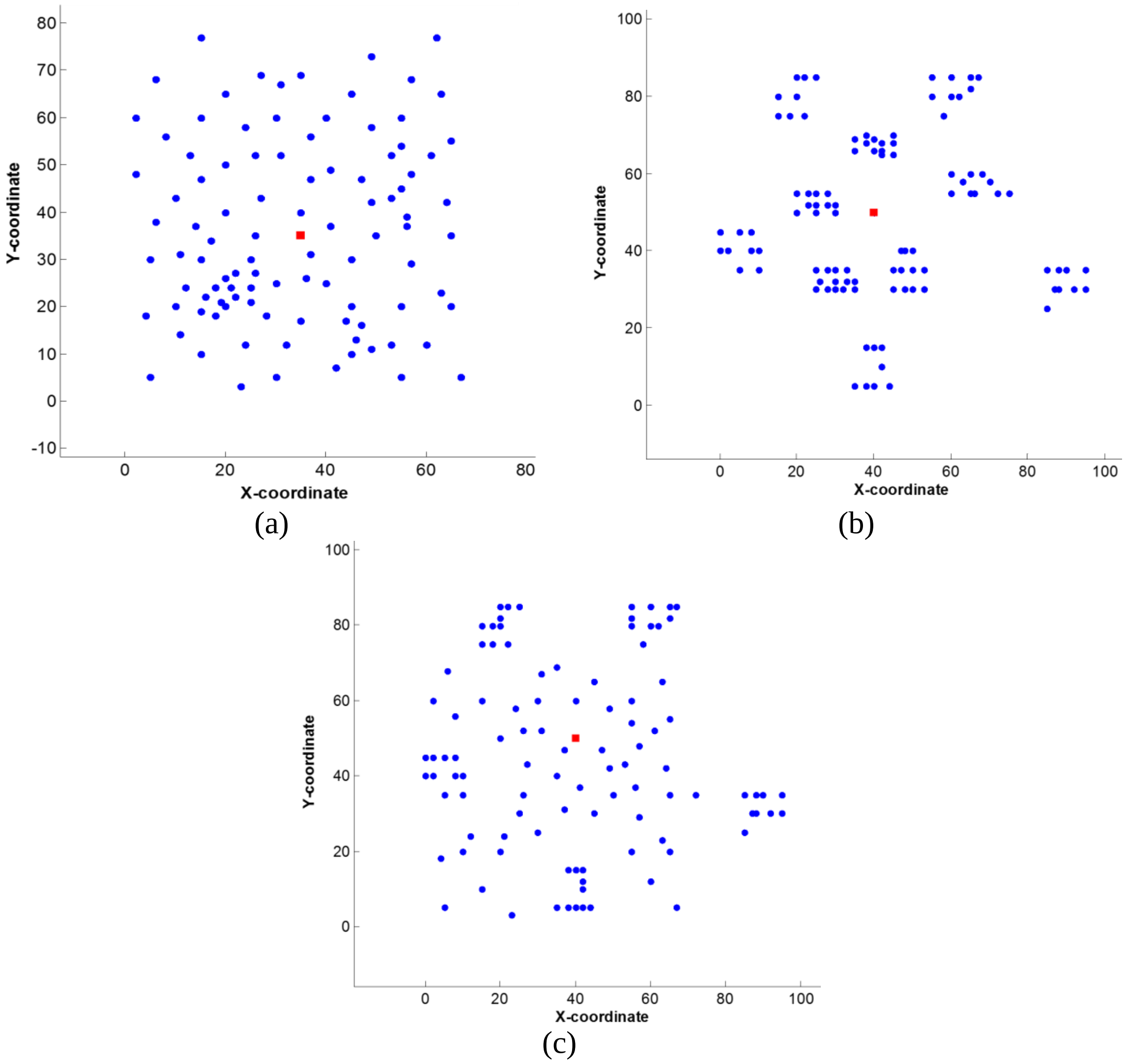


Figure 5.6. Geographical distribution patterns of Solomon benchmarks; (a) Uniform distribution (R1 and R2), (b) Clustered case (C1 and C2), (c) Semi-clustered case (RC1 and RC2); blue circles represent the customer locations and red square the depot.

Using the above benchmark instances, the experimental study has investigated the impact of a) customer *geographical distribution*, and b) customer *time-windows* on the effectiveness of the various strategies. Figure 5.6 and Figure 5.7 illustrate the patterns of these two customer attributes, respectively. We have also investigated the impact of the *degree of dynamism (dod)* (Larsen *et al.*, 2002) on strategy effectiveness. To do so, for all R1, C1 and RC1 instances we examined cases of *low dod* (25% DO), *moderate dod* (50% DO) and *high dod* (75% DO). For the MR2, MC2 and MRC2 we examined cases of *moderate dod* (50% DO), as proposed by Kontoravdis and Bard (1995).

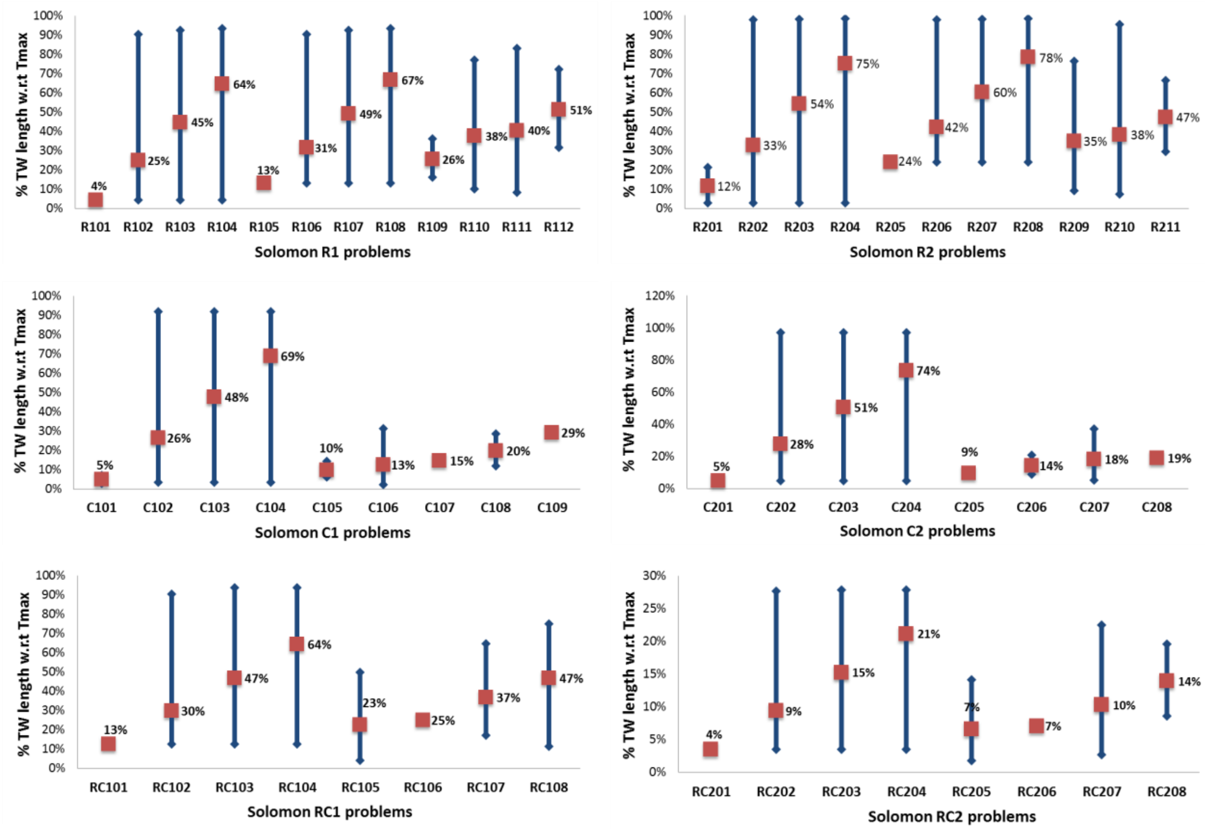
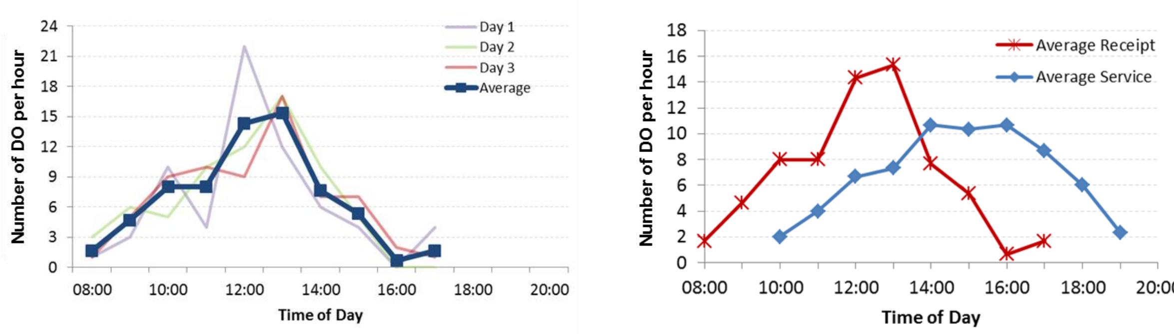


Figure 5.7. Time window information for the Solomon datasets (bars indicate the range between the min & max TW values of each instance; squares represent the average TW width)

Thus, we constructed 120 different cases (3 *dod* values for each one of the 31 test instances of R1, C1 and RC1 and 1 dod value for the 27 instances of MR2, MC2 and MRC2). For each of the 120 cases, we generated 10 different problems (different selection of static orders), resulting in a total of 1200 test problems. Note that static orders were randomly selected from each 100-customer problem and the remaining customers formed the set of DO. Also for the problems that involve TW, we skewed the selection of offline requests towards those with early TW opening. The initial solutions (original assignment of offline requests to vehicles) were obtained by a Clark and Wright savings heuristic (Clark and Wright, 1964), followed by a Reactive Tabu Search metaheuristic (Osman and Wassan, 2002) as a post-optimization process.

Finally, DO arrive during the window $[0, 0.75 * T_{max}]$ according to a continuous uniform distribution (e.g. for cases of 12 hour service period, only DO that arrive during the first 9 hours can be served). The selection of the arrival window has been motivated by a real-life courier service as illustrated in Figure 5.8a. The latter presents a) the number of requests received by the provider's call-center (operating from 08:00 to 18:00) per day, as well as the average for the three day period. The working horizon is 12 hours (from 08:00 to 20:00) and the policy of the provider is to offer service for pick-up requests at minimum one hour after the call (Figure 5.8b). The Figure clearly shows that DO are received during the first 75% of the working horizon.

For the remaining customer characteristics, i.e. on-site service times, customer demand, and the actual time-window characteristics (opening and closing times) we used the information from the original benchmark instances.



(a) (b)

Figure 5.8. (a) Number of DO per hour for a 3-day period; (b) average number of DO per hour (received and served)

5.4.2 Assessment of the heuristic B&P algorithm

In order to assess the performance of the proposed heuristic, we considered all datasets of Section 5.1.2 under the following settings: a) for datasets R1, C1, RC1, we have used all 100-node instances with 25% and 50% dod; b) for datasets MR2, MC2 and MRC2 we reduced the size of the related instances to 50-nodes by considering only the first 50 nodes of each original instance; subsequently, we generated problems with 25% and 50% dod. This reduction was necessary, since the full sized problems could not be solved to optimality by the exact algorithm within reasonable time.

For each instance, we assumed that a) all offline requests have been assigned to vehicles according to Section 5.4.1.2, and b) all DO are known before the vehicles are dispatched from the depot. Each static instance is solved by both the exact (OPT) and the heuristic (HEUR) algorithm. Note that instances R104, R108, C104, MR204, MR208 and MC204 are not included in the averages, since the OPT algorithm could not solve them within reasonable time.

Following the notation introduced in Section 5.4.1.2, Table 5.3 summarizes the results obtained for each dataset. The results have been averaged over all test problems solved per dataset (including the 10 problems per instance). The first two columns of Table 5.3 denote the dataset and the nodes per instance considered for each dataset. The subsequent column sets report the performance of HEUR and OPT for 25% and 50% *dod*. For each *dod* set, the Table reports the percentage deviation of the solution of HEUR from that of OPT *(%Dev*), and the computational times CT_{OPT} and CT_{HEUR} (in sec). The last column provides the average deviation from OPT over the two *dod*. The bottom section of the Table reports the average performance indicators per dataset. Additional indicators (distance traveled, number of routes) are provided in Table A.1 of Appendix A.

Table 5.3. Performance of heuristic B&P algorithm

Dataset	Nodes	$dod = 25\%$			$dod = 50\%$			Average %Dev
		%Dev	CT_{OPT}	CT_{HEUR}	%Dev	CT_{OPT}	CT_{HEUR}	
R1	100	2.0%	719.3	36.8	1.8%	5239.5	56.6	*1.9%*
C1	100	2.6%	136.1	24.8	2.5%	2029.0	68.6	*2.6%*
RC1	100	2.5%	188.4	32.3	2.0%	896.1	35.7	*2.3%*
MR2	50	2.1%	651.0	13.1	2.1%	6108.1	94.9	*2.1%*
MC2	50	1.4%	632.9	10.6	1.9%	3509.9	140.6	*1.7%*
MRC2	50	2.7%	382.3	8.7	2.2%	1031.3	75.5	*2.5%*
Average		2.2%	451.7	21.1	2.1%	3135.7	78.7	2.2%

Based on Table 5.3, HEUR seems to yield efficient solutions with an average deviation of 2.2% from the optimum over all datasets. Regarding the computational effort, HEUR seems to be highly efficient compared to its exact counterpart.

5.4.3 Experimental investigation of re-optimization strategies

In this Section we focus initially on the overall performance of the re-optimization strategies (Section 5.4.3.1). Subsequently, we drill down on how key parameters affect the strategy performance (Section 5.4.3.2). In Section 5.4.3.3 we investigate the quality of the solutions for the entire dynamic routing problem using the proposed re-optimization algorithms and an indicative set of instances.

For the experimental analysis, we employed all instances described in Section 5.4.1.2 and used the SRR and NRR policies. For NRR, we used $N = 0.1\widehat{N}, 0.2\widehat{N}, 0.33\widehat{N}$ (where $\widehat{N}$ is the total number of DO) hereafter designated as NRR-1, NRR-2 and NRR-3. Each policy was tested under the FR and PR release tactics, resulting to a total of eight (8) strategies for each one of the test problem (i.e. 9,600 problems in total).

It is noted that the analysis of the experimental results uses appropriate averages. The results of the re-optimization strategies for all instances have been included in Appendix A (Tables A.2-A.3). Detailed performance indicators (distance travelled and number of routes) per strategy and instance can also be found in Appendix A (Tables A.4 – A.7).

5.4.3.1 Overall performance of re-optimization strategies (tactic-policy combination)

Figure 5.9 presents the performance (w.r.t. VoI) of each re-optimization strategy (policy-tactic combination) for each investigated dataset, averaged over all instances of the related dataset and all degrees of dynamism. From this Figure, it is clear that a) the SRR-PR strategy leads to the best average performance (minimum VoI), and b) the PR tactic outperforms FR (on the average) in all datasets. The performance difference between the two tactics decreases as the number of elapsed DO per re-optimization cycle increases (less number of re-optimization cycles).

Figure 5.9 also indicates that the performance of each tactic is related to the frequency of re-optimization. The PR tactic seems to be more efficient for shorter re-optimization cycles. Possible causes for this include: i) short route portions have been completed when re-optimization is applied, allowing for more options, and ii) DO that are not planned for service until the next re-optimization timestamp are reconsidered, providing more possibilities for DO combinations. This behavior seems to be consistent for all investigated datasets.

The FR tactic seems to be less efficient when re-optimization is applied very frequently (SRR) for all datasets or infrequently (NRR-3) for datasets R1, C1, RC1. A possible cause for the former may be that frequent re-optimization (i.e. upon the arrival of every request) is greedy, not taking advantage of combinations of newly arrived DO. In the case of infrequent re-optimization, a larger portion of the route has been completed and fewer options are available for incorporating the newly arrived DO.

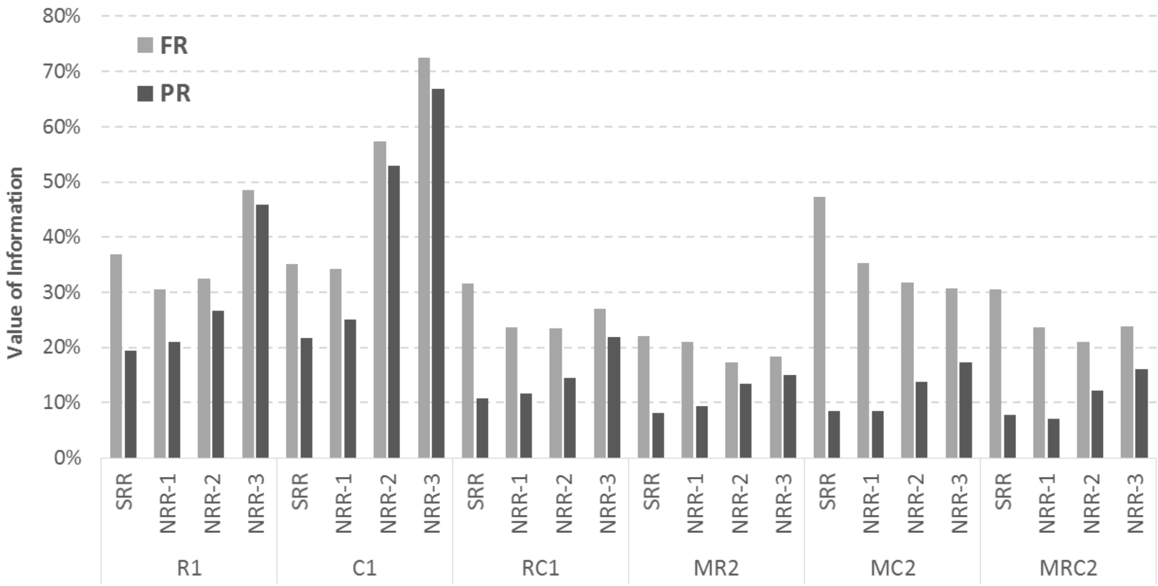


Figure 5.9. Average performance of re-optimization strategies for the different datasets

Table 5.4 reports the computational times (per re-optimization cycle) for all re-optimization strategies averaged over all test instances per dataset for 50% *dod*. The computational effort is reported in seconds and corresponds to the average running time of the heuristic during each re-optimization cycle. As expected, PR requires more computational effort. Regarding policies, again as expected, the computational effort increases as the re-optimization frequency decreases. Furthermore, the MR2, MC2, MRC2 problems (with increased horizons) seem to be more demanding. In general, the results indicate the efficiency of the proposed heuristic, since the average computational time under the FR tactic is less than 20 sec, while under the PR tactic less than 1 minute. Similar trends are valid for the 25% and 75% dod cases; the 75% dod case requires about twice the effort of the 50% dod case.

Table 5.4. Average computational effort of re-optimization strategies per dataset for 50% dod

Dataset	FR				PR			
	SRR	**NRR-1**	**NRR-2**	**NRR-3**	**SRR**	**NRR-1**	**NRR-2**	**NRR-3**
R1	0.1	0.6	2.1	10.7	3.6	6.9	13.1	20.2
C1	0.1	0.5	1.4	8.4	2.3	3.6	4.5	19.0
RC1	0.2	0.8	2.5	7.2	7.4	10.0	12.7	18.0
MR2	0.1	1.2	6.5	17.5	16.9	19.5	25.4	32.5
MC2	0.2	1.1	4.3	19.3	21.4	37.7	44.2	49.9
MRC2	0.1	1.0	4.1	20.2	19.7	23.4	25.6	35.5

5.4.3.2 Performance of re-optimization strategies under various conditions

Below we drill down to the interactions of the various re-optimization strategies with three (3) key parameters: i) TW pattern, ii) urgency of DO w.r.t. the requested service, and c) the degree of dynamism (dod).

First we investigate how the strategies perform for various TW patterns (the latter characterized by the ratio of the average TW width of all customers in the instance over. T_{max}). To do so, we grouped all investigated instances in four categories, as shown in Table 5.5. TW-1 group comprises instances with relatively narrow TW, while TW-4 comprises instances with wide or no TW.

Table 5.5. Classification of instances in TW-pattern groups

Group	% of T_{max}	# Instances	Instances
TW-1	<15%	17	R101, R105, C101, C105, C106, C107, RC101, R201, C201, C205, C206, RC201, RC202, RC205, RC206, RC207, RC208
TW-2	15% - 30%	14	R102, R109, C102, C108, C109, RC102, RC105, RC106, R205, C202, C207, C208, RC203, RC204
TW-3	30% - 50%	14	R103, R106, R107, R110, R111, C103, RC103, RC107, RC108, R202, R206, R209, R210, R211
TW-4	>50%	11	R104, R108, R112, R100,C104, C100, RC104, R203, R204, R207, R208, C203, C204

Figure 5.10 presents the average performance of all strategies for the aforementioned TW-pattern groups. For the PR tactic, all policies seem to follow similar behavior; that is, more frequent re-optimization (SRR and NRR-1) yields better results in almost all cases, irrespective of the TW width, reflecting the fact that frequent re-optimization may allow re-allocation of orders in a more flexible manner. The performance of SRR under PR appears slightly inferior to NRR-1 for instances with average TW width more than 30% of T_{max}. For the FR case, frequent re-optimization seems to favor solution quality for tight TW cases (TW-1 category). Medium-interval re-optimization cycles (NRR-1 and NRR-2) display more consistent behavior for all TW-pattern groups.

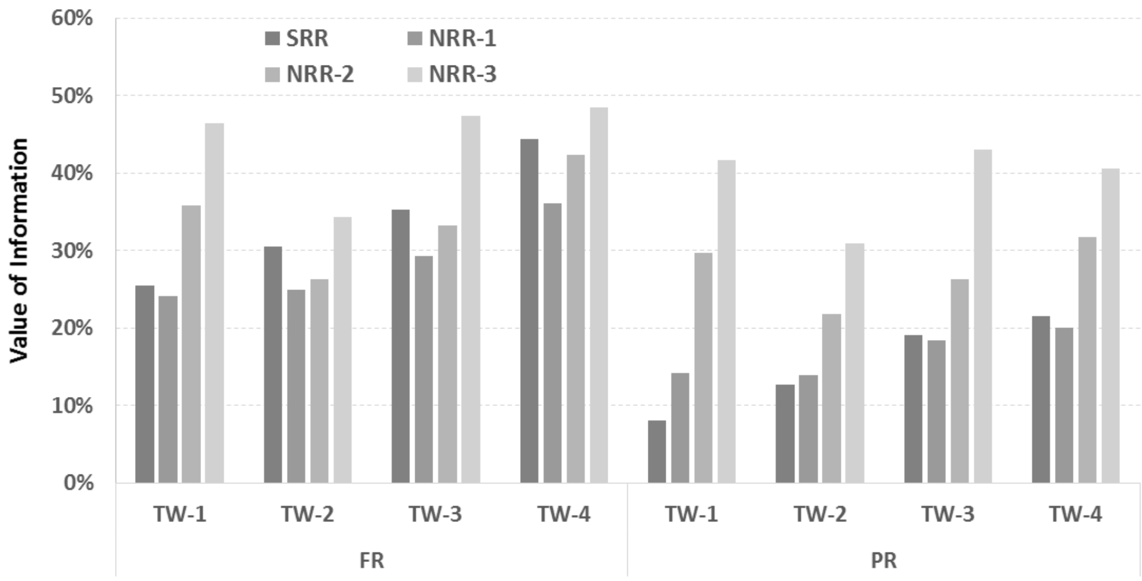


Figure 5.10. Average performance of re-optimization strategies for different TW pattern groups

To study the behavior of the re-optimization strategies for various levels of urgency w.r.t. TW closing, we have used the *effective degree of dynamism* ($edod$), as defined by Larsen (2002). The $edod$ considers the *reaction time*, i.e. the difference between the closing time b_i of the TW and the arrival time h_i of request i; longer reaction times provide higher flexibility in inserting a request in the current plan. Denoting as $\bar{N}$ the set of customers of an instance, $edod$ is defined as:

$$edod = \frac{1}{|\bar{N}|}\sum_{i\in\bar{N}}\left(1-\frac{b_i - h_i}{T_{max}}\right) \tag{5.2}$$

We have grouped all experimental instances into four levels according to the average $edod$ of all test problems (10 replicates) in each instance. The grouping is shown in Table 5.6 where *edod-1* comprises instances of high urgency (limited reaction time), while *edod-4* comprises instances with low urgency.

Table 5.6. Classification of instances in different $edod$ levels

Level	$edod$	# Instances	Instances
edod-1	>65%	12	R101, R105, C101, C105, RC101, RC105, R201, R205, C201, C205, RC201, RC205
edod-2	55% - 65%	18	R102, R106, R109, R110, C102, C106, C107, C109, RC102, RC106, R202, R209, C206, C207, C208, RC202, RC206, RC207
edod-3	45% - 55%	13	R103, R107, R111, C108, RC103, RC107, RC108, R206, R210, R211, C202, RC203, RC208
edod-4	30% - 45%	13	R104, R108, R112, C103, C104, RC104, R203, R204, R207, R208, C203, C204, RC204

Figure 5.11 presents the average performance of re-optimization strategies for the various *edod* levels. For the PR tactic, similarly to the TW-pattern analysis, frequent re-optimization (SRR and NRR-1) yields better results across all *edod* levels, while infrequent re-optimization deteriorates the solution's performance. Furthermore, it seems that in cases of low urgency (*edod-4),* NRR-1 is very competitive and performs even better than SRR. Regarding the FR tactic, medium-interval re-optimization policies (NRR-1 and NRR-2) seem to favor the solution quality for all $edod$ levels.

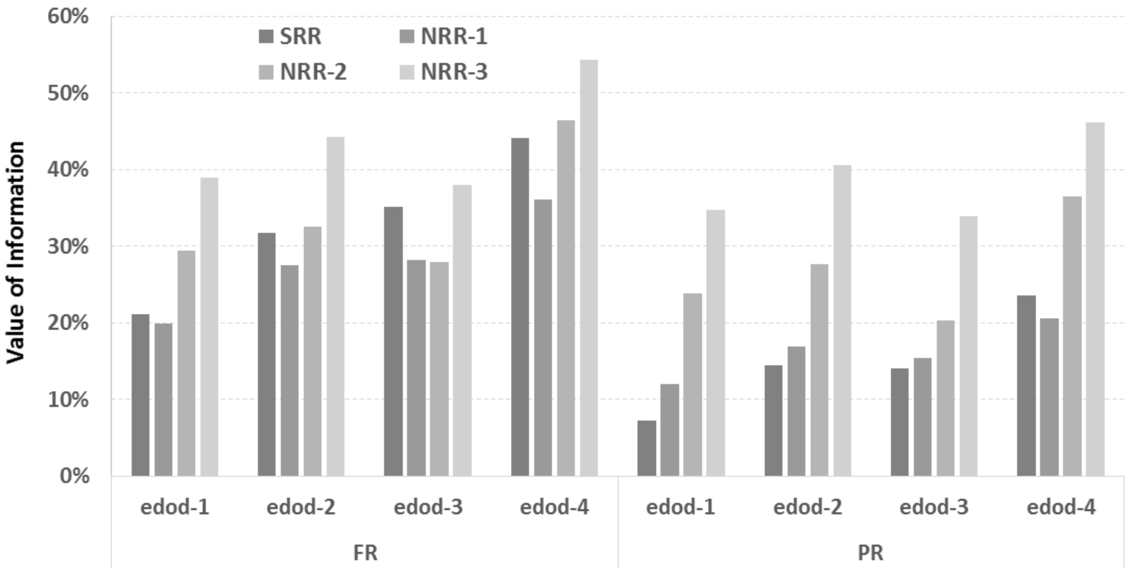


Figure 5.11. Average performance of re-optimization strategies for different $edod$ levels

In terms of the interaction of the re-optimization strategies with the degree of dynamism (*dod*) and the customer geographical distribution, Figures 5.12 and 5.13 present the performance of each policy w.r.t. *dod* for the PR and FR tactics, respectively, and for different geographical distributions (R1, C1 and RC1). Note that this is the average performance over all related instances.

For the PR tactic (Fig. 5.12), SRR and NRR-1 outperform all other policies for all cases of *dod*, while the performance deteriorates with increasing degree of dynamism across all policies. In environments with strong dynamism, many vehicles are dispatched from the depot to handle the increased DO numbers. This causes additional non-productive costs (travel to/from depot). Infrequent re-optimization in such cases causes vehicles *en route* to return to the depot at an early stage (because of the limited number of committed orders) and new vehicles to be dispatched in order to cover the high demand. For FR (Fig. 5.13), the NRR-1 (and partially NRR-2) policy yield improved solution quality across *dod*, more so in cases of strong *dod*.

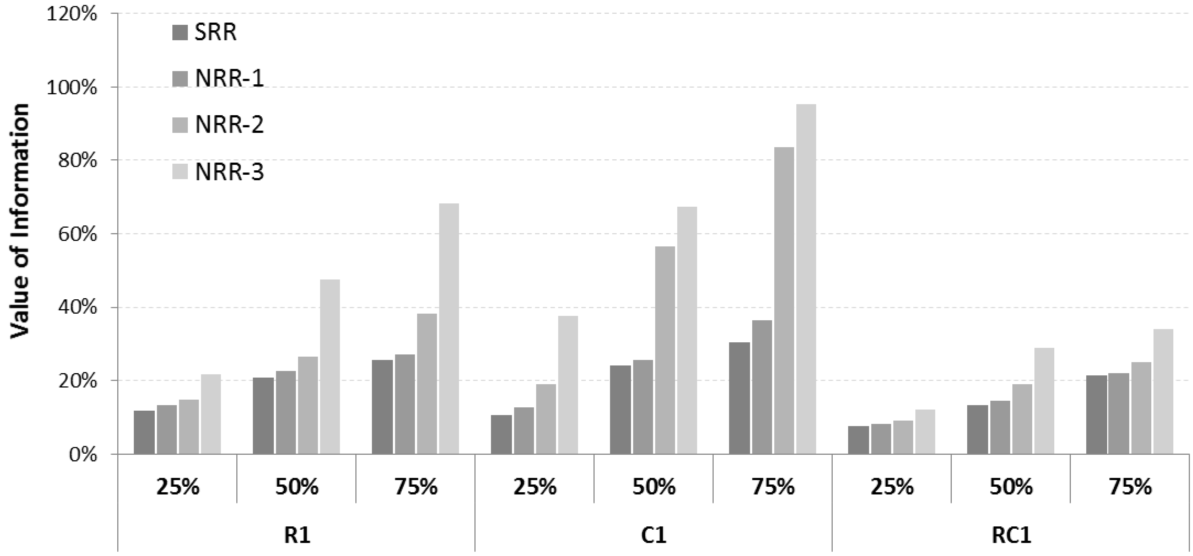


Figure 5.12. Performance of policies under the **PR** tactic for various geographical distributions and dod levels

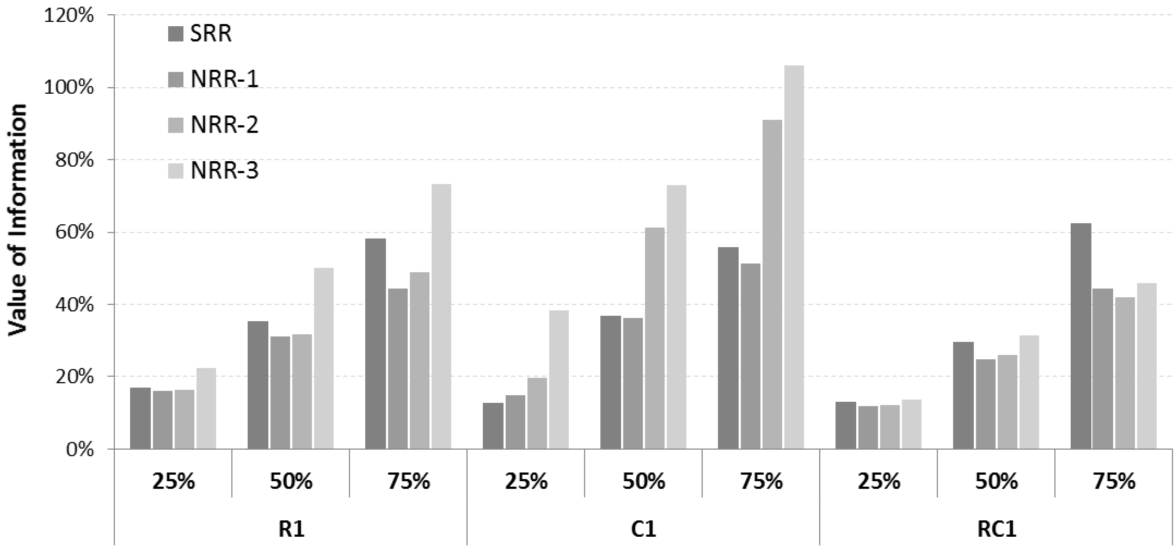


Figure 5.13. Performance of policies under **FR** tactic for various geographical distributions and dod levels

Finally, the related Figures indicate that policies present similar relative behavior for different values of *dod* for the R, C and RC configurations (no strong interaction). For the C configuration, the solution quality deteriorates significantly when infrequent re-optimization is used (NRR-2 and NRR-3), especially for cases with medium to strong dynamism. This may be caused by the fact that a large portion of the route corresponds to travelling back to depot; since no diversion is allowed, vehicles *en route* are not considered as available during their return trip and, thus, re-optimization tends to use more vehicles located at the depot to serve DO.

5.4.3.3 Performance of re-optimization strategies w.r.t. the algorithm employed

It is worth investigating whether finding an optimal solution during each re-optimization cycle leads to superior solutions for the entire dynamic problem. For example, one may suspect that such a practice may lead to locally aggressive optimization that allows new DO to be incorporated into the current plan at a significant detour cost.

To investigate the effect of the re-optimization algorithm, we solved using both OPT and HEUR a series of selected problems (selected R1 instances with 25% and 50% *dod*), for which the exact B&P approach is applicable. For those problems we applied the SRR, NRR-1 and NRR-2 policies under the PR tactic. Figure 5.14 illustrates the results obtained averaged over the two *dod* values (25% and 50%). The results have been reported as the difference between VoI provided by HEUR minus the VoI provided by OPT ($V_F(HEUR) - V_F(OPT)$); thus, positive values reflect superiority of OPT results over HEUR.

The results are divided among OPT and HEUR. The positive effect of HEUR over OPT to the overall problem seems to concern cases with wide TW of varying width values (e.g. R103, R106, R110). On the other hand, OPT seems to perform better for cases with consistent TW patterns (low or zero variance). These cases in general provide limited options for incorporating DO in the routing plan; hence, inferior quality solutions during early re-optimization cycles may not provide better results on later re-optimization cycles.

Overall the suspicion that aggressive re-optimization may in some cases lead to inferior overall solutions has been confirmed.

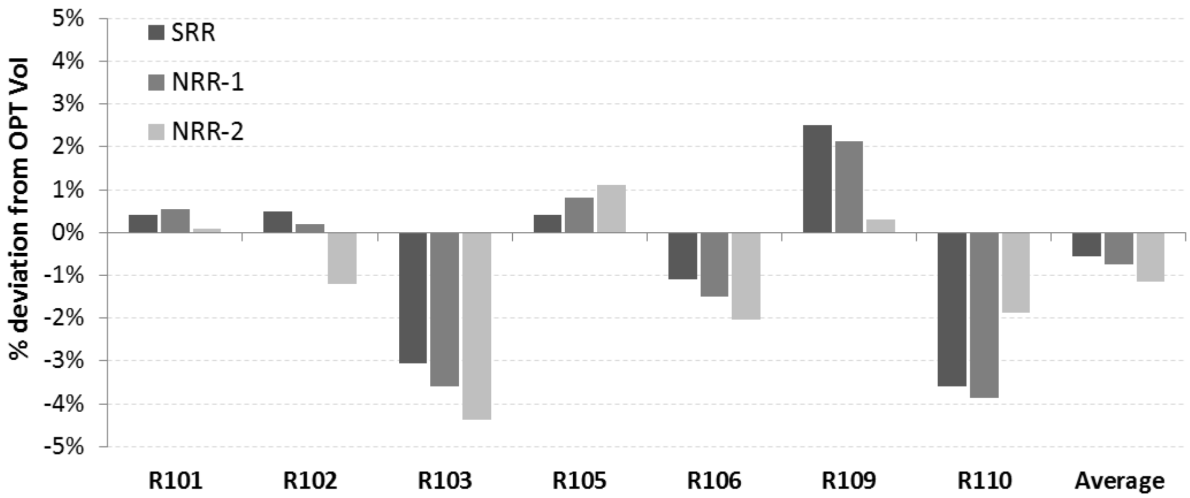


Figure 5.14. Comparison of the effects OPT and HEUR

5.5 Concluding remarks

In this Chapter, we drilled-down to significant aspects concerning the re-optimization process, i.e. *when* to re-optimize, and *what* part of the new plan to communicate to the drivers. We presented and analyzed typical re-optimization policies found in practice, i.e.: i) re-optimization upon the arrival of each DO, ii) re-optimization after a certain number of DO have been received. In addition, we investigated the effect of two implementation tactics: i) immediate release of all DO for implementation (FR) and, ii) release of only those DO that are scheduled for implementation prior to the next re-optimization cycle (PR). We provided theoretical insights regarding the expected behavior of those tactics and we illustrated through extensive experimentation that re-optimization upon the arrival of each DO under the PR tactic provides superior results on the average. However, this policy seems to be the least favorite, when the FR tactic is employed.

Furthermore, we assessed the performance of the re-optimization strategies under various operating scenarios. Our experimentation has illustrated the following:

i) When the business case allows it, one should always re-optimize under the PR tactic in as short re-optimization intervals as possible

ii) When the FR tactic is unavoidable due to the characteristics of the practical environment, one should prefer re-optimization over short to medium intervals for cases of tight to medium TW, and over medium to larger intervals for wider TW cases

iii) In environments with strong dynamism, medium interval policies (regardless of tactic) seem to provide the safest option.

Table 5.7 summarizes the aforementioned results. Considering that PR is superior to FR, the Table presents the best possible re-optimization option under each tactic as it emerged from the experimental study, w.r.t. to the problem attributes. For simplicity, we included two (2) options for the NRR policy; medium and long re-optimization periods.

Table 5.7. Preferable re-optimization policies per parameter

Parameter	Description	FR				PR			
		SRR	NRR *(medium)*	NRR *(long)*	FTR	SRR	NRR *(medium)*	NRR *(long)*	FTR
Geographical distribution	Uniform		✓			✓			
	Clustered		✓				✓		
Time Windows	Tight		✓			✓			
	Medium/Wide		✓			✓			
	Very wide		✓				✓		
	No TW		✓			✓			
DoD	Weak		✓			✓			
	Moderate		✓			✓			
	Strong		✓				✓		

Chapter 6: THE DVRPMB FOR THE CASE OF LIMITED RESOURCES

In Chapter 5 we studied the re-optimization problem by assuming an unlimited vehicle fleet available to serve all (static and dynamic) orders. This allowed us to investigate the performance of re-optimization strategies under a single objective, i.e. minimize distance traveled. In this Chapter, we examine the case of DVRPMB, in which the number of available vehicles is limited. To do so, we introduce appropriate objective functions that account for vehicle productivity during each re-optimization cycle and we illustrate that these objectives can offer higher customer service compared to conventional ones that account strictly for either cost minimization or service maximization.

Section 6.1 presents significant considerations related to the constraint of limited resources, along with a review of the relevant literature. Section 6.2 discusses the objective functions for DVRPMB with limited resources and sets the related theoretical foundation. Section 6.3 presents the necessary modifications of the branch-and-price algorithm of Chapter 4 to address the re-optimization problem. Section 6.4 investigates experimentally the effect of the proposed objective functions on the efficiency of the solution under different problem settings. Finally, Section 6.5 describes the application of the proposed method in a practical courier environment.

6.1 Introduction and background

We refer to the case of DVRPMB with limited available vehicles as the *DVRPMB with limited resources* (*m*-DVRPMB). In this case, due to the resource constraints, some of the newly received dynamic orders (DO) may not be served, raising some interesting considerations discussed below.

The first of these considerations relates to the objective function of the re-optimization problem. As we will demonstrate below, conventional objectives that account strictly for either cost minimization or service maximization are either inappropriate, or may not be adequate to address the problem effectively. In the case of cost minimization, and since the constraint of serving all orders is not enforced, dynamic orders increase cost and, thus, are left unserved. On the other hand, in the service maximization case, in an effort to include as many orders as possible in the current re-optimization cycle costs may increase significantly, which may lead to serving less DO eventually.

A related consideration in this dynamic setting concerns the prioritization of clients at each re-optimization cycle. Specifically, during a certain re-optimization cycle it may be beneficial to favor the service of certain customer orders (e.g. urgent ones) in the expense of others, under the assumption that the excluded (e.g. not urgent) ones can fit in the plan during a subsequent re-optimization cycle. Thus, one should examine whether it is beneficial to prioritize service of certain orders, and if so, under which conditions this is favorable to the problem's objective. This consideration is even more important in a dynamic, deterministic environment, in which no forecasting information is available.

Despite the practical importance of *m*-DVRPMB, the problem has not been addressed in the literature. If all orders to be served in the future would be known in advance, the *m*-DVRPMB would reduce to the Pickup and Delivery Problem (PDP) with Selective Pickups (Sural and Bookbinder, 2003; Gribkovskaia *et al*, 2008; Gutiérrez-Jarpa *et al.*, 2010). In this very interesting and practical problem, all deliveries must be performed but pick-ups are optional; however, pick-ups generate a profit when fully collected (i.e. partial pick-ups are not allowed). The objective is to minimize the routing cost minus the collected revenue. This problem arises naturally in reverse logistics contexts, in which customers return goods (e.g. empties) to the depot.

The PDP with Selective Pickups was first addressed by Sural and Bookbinder (2003) for a single vehicle. The authors used an exact branch-and-bound technique to solve it by employing flow variables and adaptations of the well-known Miller–Tucker–Zemlin constraints (Miller *et al.*, 1960) in order to prevent subtours. Their approach was able to solve instances of up to 30 customers. More recently, Gutiérrez-Jarpa *et al.* (2009) proposed a branch-and-cut algorithm, able to solve instances of up to 90 customers. Prive *et al.* (2006) developed heuristic methods to study a practical problem, which involved the delivery of soft drinks to convenience stores in the city of Quebec, and the collection of empties (bottles or cans) using a heterogeneous fleet of vehicles. In their problem formulation, pick-ups were associated with revenue and they could be performed only if they didn't violate vehicle capacity constraints.

Aas *et al.* (2007) studied the routing of supply vessels to offshore installations, in which vessels pick-up empty containers (or waste). In this problem, due to limited capacity, it is not always possible to serve all collections and, thus, priority is given to the most important ones. The authors formulated the problem as a mixed integer linear program, and they were able to solve practical instances with 10 installations to optimality using CPLEX 9.0. Gribkovskaia *et al.* (2008) studied a similar application and proposed a tabu search method to the single vehicle pick-up and delivery problem with selective pick-ups.

More recently, Gutiérrez-Jarpa *et al.* (2010) developed an exact branch-and-price algorithm for the Vehicle Routing Problem with Deliveries, Selective Pickups and Time Windows (VRPDSPTW). The authors categorized the customer requests in: i) pick-up and delivery requests that are disjoint (P/D) and ii) pick-up and delivery operations associated with the same customer (P&D). Based on this classification, they addressed five variants of the VRPDSPTW, i.e. i) P/D problems, in which customers may be served in an arbitrary order, ii) P/D problems with backhauls (i.e. all pick-ups must be performed after the deliveries), iii) P&D problems, in which each customer can be visited exactly once, iv) P&D problems that allow multiple visits to the same customer, and customers can be visited in an arbitrary order, and v) P&D problems with multiple visits and backhauls. The authors were able to solve instances of up to 50 customers to optimality.

The static version of the problem investigated in this chapter would be similar to the one studied in the above references, if all pick-up (collection) orders had equal or zero profit (therefore, in effect, the profit gained from including a pick-up order in the plan would not be dependent on the routing costs). We differentiate our work in the following two aspects; first, to the best of our knowledge, no other study has focused on the dynamic version of this problem. Second, to deal with dynamism and leverage on the opportunity offered by the multiple re-optimization cycles, we propose to consider vehicle productivity in the re-optimization process, i.e. to maximize vehicle utilization within an appropriate period of time in anticipation of additional dynamic orders.

To address m-DVRPMB, we extend the branch-and-price (B&P) approach presented in Chapter 4 for solving the related re-optimization problem. To do so, we study alternative objective functions that maximize service, while, at the same time, enhance vehicle productivity. Both the original formulation of DVRPMB and the solution procedure (Master Problem and Subproblems) have been modified appropriately. We evaluate the performance of the proposed objective functions with respect to a conventional, but relevant, objective function that accounts only for service maximization, under various operating scenarios and parameters. Finally, we apply our proposed method to a case study of a next-day courier service provider.

6.2 Objective functions for the m-DVRPMB

We introduce three objective functions to deal with limited resources in the context of the re-optimization problem: a) a conventional one that maximizes service by assigning a fixed profit to each DO served (objective $\check{z}_1$) and b) a proposed objective function that provides additional profit for each order (static or dynamic) served within the next (upcoming) re-optimization cycle (objective $\check{z}_2$), and c) an objective that modifies $\check{z}_2$ in terms of the additional profit term; in this case, the profit concerns all orders to be served at any future period and it decreases linearly depending on the period (re-optimization cycle) the order is served (objective $\check{z}_3$).

Let ξ_u denote the fixed profit assigned to each served DO and ξ_p the additional profit in case an order (static or dynamic) is served within the upcoming re-optimization cycle; thus, the profit per order corresponding to the three (3) alternative objective functions varies as illustrated in Figure 6.1. By using the appropriate function, we may steer the solution method into maximizing customer service (objective $\check{z}_1$), as well as maximizing vehicle productivity (objectives $\check{z}_2$ and $\check{z}_3$). Sections 6.2.1 and 6.2.2 describe the structure of the proposed objective functions; Section 6.2.3 discusses some fundamental aspects regarding the expected performance of these objectives.

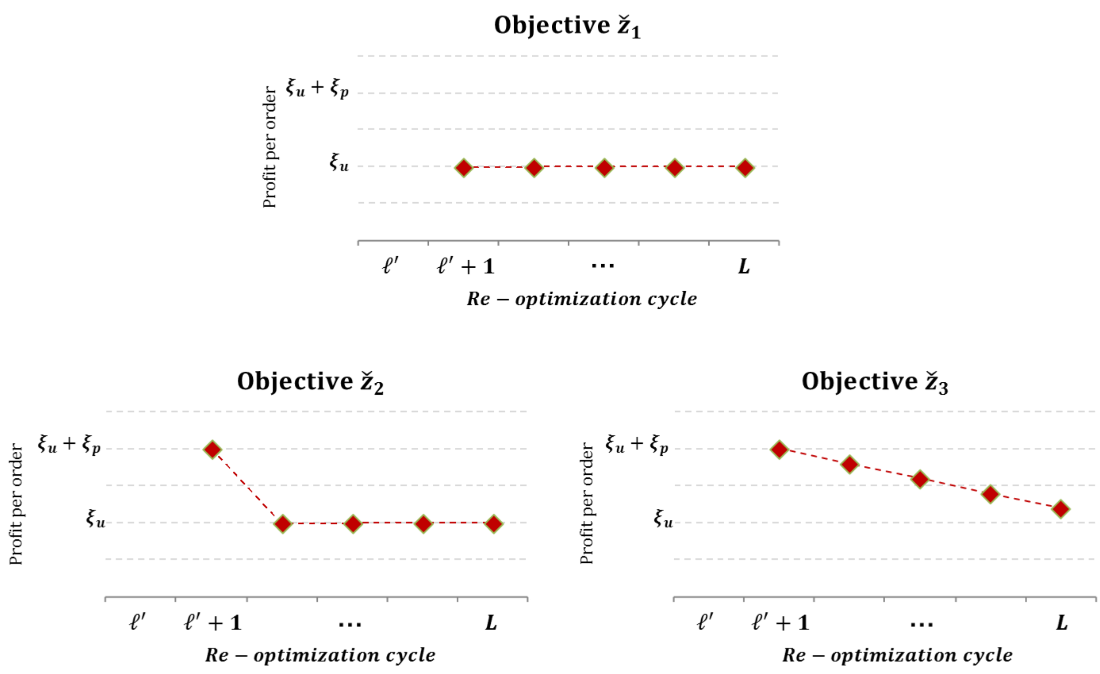


Figure 6.1. The profit per order according to the three objective functions employed

For objectives $\check{z}_2$ and $\check{z}_3$ to be relevant, the re-optimization time instances have to be predetermined (known *a priori*). The Fixed-Time Re-optimization (FTR) policies discussed in Chapter 5 (see Section 5.2) are appropriate in this case. Re-optimization policies that depend on the number of arrived DO (e.g. SRR and NRR policies) may be only implemented under objective $\check{z}_1$.

6.2.1 A conventional objective function that maximizes service

Recall that in the re-optimization problem of the typical DVRPMB all customers may be served (i.e. there are enough resources to serve all customer orders). Thus, the objective function of DVRPMB (as stated in Chapters 3 and 4) strictly minimizes the routing cost. Under this objective, if the constraint for serving all orders is relaxed in the re-optimization problem, then (in general) no dynamic order will be included in the final solution, since serving it will increase the routing cost. To address this issue, one can introduce additional (profit) terms in the objective function in order to simultaneously:

(a) Increase the number of DO (set F) served throughout the remaining horizon – primary objective

(b) Decrease the total cumulative routing costs (over the remaining horizon) – secondary objective.

That is,

$$\min(\check{z}_1) = -\xi_u \sum_{k \in V} \sum_{(i,j) \in A | i \in F, j \in W} x_{ijk} + \sum_{k \in V} \sum_{(i,j) \in A} c_{ijk} x_{ijk} \quad (6.1)$$

Under an appropriately large positive value of ξ_u, this objective maximizes the number of served DO, and among the solutions with maximal number of served DO, selects the one with minimum routing cost. Determining this value is straightforward: consider a re-optimization problem in which an additional DO is to be included in the current plan. If the routing cost for incorporating this DO is higher than ξ_u, then the solution will not incorporate this DO, since the overall objective value will increase. Thus, ξ_u should be set to a value that exceeds the upper bound of the cost (worst case) of incorporating a DO in the current plan. This can be achieved by setting ξ_u to a value larger than $\max_{i \in F}(c_{r_i})$, where c_{r_i} represents the cost of the unit route $[Depot - i - Depot], \forall i \in F$.

This straightforward objective may be appropriate for a static planning problem with limited resources (for which all orders are known *a priori*), but may not be adequate in the setting under study, since additional orders are expected to arrive. The anticipation of additional work favors reserving fleet capacity for the latter periods of the operational horizon so that newly arriving DO may be served. This, in turn, indicates that the available fleet should complete as much of the known work as early as possible (i.e. increase the productivity of the system in the early re-optimization cycles), in order to reserve capacity for the later re-optimization cycles.

Specifically, there are multiple ways that fleet productivity, and thus the capacity of the system to serve new DO, may be impacted adversely by objective $\check{z}_1$, especially during early re-optimization cycles, in which the few DO known up to that point in time may all fit in the plan. In these early cycles using $\check{z}_1$:

i. May cause the incorporation of DO at a significant detour cost (as also discussed and illustrated experimentally in Section 5.4.3.3)
ii. Some vehicles may be forced to wait for long periods at customer sites for a TW to open (since it may be more cost-efficient to assign a DO with late opening time to a vehicle closer to it – see also example in Figure 6.2)
iii. Vehicles stationed at the depot may not be used, since it might be more cost-efficient to assign a DO to a vehicle already *en route*.

All these potentialities may decrease significantly the productivity of the fleet.

However, neglecting entirely the routing costs in an attempt to increase the productivity of the fleet is also not appropriate, since an excessive increase of costs in the current re-optimization cycle, might decrease the productivity of the vehicles in the subsequent (future) re-optimization cycles.

Below we enhance the objective function in order to maximize productivity of the fleet appropriately in the upcoming re-optimization cycle(s), in anticipation of additional work to come, without excessively compromising the capacity of the system at later re-optimization cycles.

6.2.2 A proposed objective function that accounts for vehicle productivity

In addition to maximizing the total number of DO served, we propose an enhanced objective function, which attempts to maximize the number of orders served within the upcoming re-optimization cycle (the length of which is known in advance); it does so, however, among the solutions with the same number of DO served. Consider the re-optimization problem during the ℓ-th cycle, and let ω_{ik} denote a decision variable that is equal to 1 if order $i \in N$ ($N = C \cup F$ are all orders involved in the re-optimization cycle) is served during the time interval $[T_\ell, T_{\ell+1}]$ by vehicle $k \in V$ and 0 otherwise. Then, the proposed objective function (denoted as $\check{z}_2$) seeks the following in lexicographical order:

(a) Maximize the number of dynamic orders (F) served throughout the remaining horizon

(b) Maximize the number of both static and dynamic orders (N) served within the upcoming re-optimization cycle (i.e. within time interval $[T_\ell, T_{\ell+1}]$)

(c) Minimize the routing cost

$$\min(\check{z}_2) = -\xi_u \sum_{k \in V} \sum_{(i,j) \in A | i \in F, j \in W} x_{ijk} - \xi_p \sum_{k \in V} \sum_{i \in N} \omega_{ik} + \xi_c \sum_{k \in V} \sum_{(i,j) \in A} c_{ijk} x_{ijk} \quad (6.2)$$

where ξ_p corresponds to a positive value (profit) if an order is served within the time interval $[T_\ell, T_{\ell+1}]$ (of known duration).

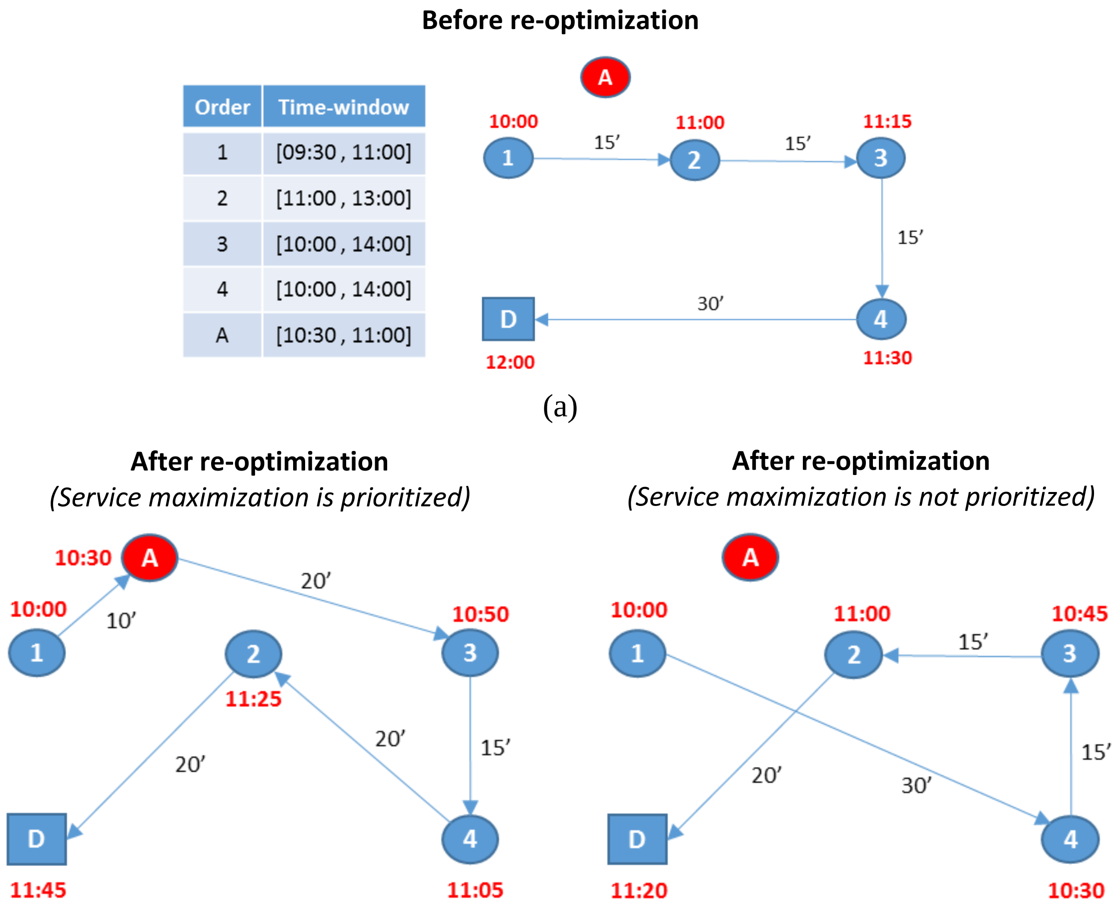
Although the primary goal of objective $\check{z}_2$ is to maximize the DO served, the purpose of term (b) in the objective function is to maximize the productivity of the fleet during the current cycle in anticipation of additional work to come; i.e. with this term we attempt to encourage the deployment of resources as early as possible (during early re-optimization cycles), even if this results in higher routing costs in the solution of the current re-optimization problem. Note that objective $\check{z}_2$ attempts to maximize the productivity of the fleet in the current re-optimization cycle by favoring all N orders (both static and dynamic) to be served within time interval $[T_\ell, T_{\ell+1}]$.

Prioritizing term (a) over term (b) in the objective function is important in such dynamic context, since it is uncertain if a DO not included in the solution of the current re-optimization cycle can be served at a later stage. For solutions under the FR tactic, this is obvious, since DO that do not fit in the solution of the current re-optimization cycle are not considered during future ones. However, for solutions under the PR tactic, in which DO not included in the solution of the current cycle are re-considered, we distinguish two relevant scenarios:

i) In cases in which all DO can be served by the solution of the current re-optimization cycle, term (a) of the objective is not important, and term (b) guides the solution to serve as many orders as possible within the interval $[T_\ell, T_{\ell+1}]$. In these cases term (b) then will avoid the incorporation of DO in the expense of significant resources (e.g. cost/time).

ii) In cases in which not all DO fit in the plan, then prioritizing term (a) over (b) ensures that the objective will not force the service of more orders (static and dynamic) during the interval $[T_\ell, T_{\ell+1}]$ at the expense of incorporating a DO in the current plan.

To better illustrate the latter case, consider the example of Figure 6.2a, which presents the state of a single route at the re-optimization time instance $T_1 = 10{:}00$. At time T_1, the vehicle is located at customer 1 and is scheduled to serve three static orders (2, 3 and 4), while dynamic order A needs to be incorporated in the current plan (with time window opening at $10{:}30$). The expected time of arrival to each order (prior to re-optimization and incorporation of order A), as well as the travel duration of each planned arc are also displayed in the Figure (note that in this state, the vehicle arrives at customer 1 at 10:00, but waits for 45 minutes, due to the opening of the time window of customer 2). Assume also that $T_2 = 11{:}00$. If we prioritize term (a) over term (b) in this example, the objective will attempt to maximize the number of orders served within time interval $[T_1, T_2]$ among the solutions that incorporate order A. The result of this scenario is illustrated in Fig. 6.2b, in which order A has been incorporated in the plan and one static order (order 3) has been also served till the next re-optimization (till $11{:}00$). Fig. 6.2c illustrates the opposite, i.e. when term (a) is not prioritized over term (b). It is clear from Fig. 6.2c that dynamic order A will remain unplanned, since due to the objective, the preferred solution is the one that serves the three static orders (2, 3 and 4) during the interval [10:00,11:00] (i.e. one more order will be served during this interval compared to solution of Fig. 6.2b). Order A may not be served since its time window has elapsed.



Order	Time-window
1	[09:30 , 11:00]
2	[11:00 , 13:00]
3	[10:00 , 14:00]
4	[10:00 , 14:00]
A	[10:30 , 11:00]

(a)

(b) (c)

Figure 6.2. Example for comparing between alternative expressions of objective $\check{\boldsymbol{z}}_2$

Thus, we choose to maintain the primacy of term (a) of the objective over term (b). To do so, the following should hold:

$$\xi_u > \xi_p \sum_{k \in V} \sum_{i \in N} \omega_{ik} \tag{6.3}$$

Note that the largest possible value of the right hand side of (6.3) is obtained if all remaining orders (of set N) are served in the next re-optimization cycle (i.e. $\sum_{k \in V} \sum_{i \in F} \omega_{ik} \ = |N|$). Thus, Ineq. (6.3) can be written as:

$$\xi_u > \xi_p * |N| \ \Rightarrow \ \xi_p < \frac{\xi_u}{|N|} \tag{6.4}$$

This may be satisfied with:

$$\xi_p = \frac{\xi_u}{|N| + 1} \tag{6.5}$$

Working along the same lines, in order to guarantee the primacy of term (b) over term (c), the value of ξ_p should be larger than the largest possible value of the routing cost, i.e.:

$$\xi_p > \xi_c \sum_{k \in V} \sum_{(i,j) \in A} c_{ijk} x_{ijk} \tag{6.6}$$

The largest possible value of the routing cost (upper bound) can be assumed when all N orders are included in the solution and they are served directly from the depot, since each order $i \in N$ can be served exactly once and the fleet is homogeneous. Thus, Ineq. (6.6) can be reduced to:

$$\xi_p > \xi_c \sum_{k \in V} \sum_{j \in N} c_{0jk} \tag{6.7}$$

Assuming $\check{C} = \sum_{k \in V} \sum_{j \in N} c_{0jk}$ and replacing ξ_p based on Ineq. (6.5), we have:

$$\xi_p > \xi_c * \check{C} \xRightarrow{\xi_p = \frac{\xi_u}{|N|+1}} \frac{\xi_u}{|N|+1} > \xi_c * \check{C} \tag{6.8}$$

This may be satisfied by:

$$\xi_c = \frac{\xi_u}{(|N|+1) * \check{C} + 1} \tag{6.9}$$

Using in $\check{z}_2$ the values of Eqs. (6.5) and (6.9) will ensure that from those solutions that maximize the number of served DO, the one to be selected a) serves as many orders as possible in the interval $(T_\ell, T_{\ell+1}]$, and b) has the minimum routing cost among the ones serving the same number of orders in this interval.

As already discussed, forcing as many orders as possible to be served until time $T_{\ell+1}$ may cause higher routing costs, compared to scenarios in which orders may be served at an appropriate time. Objective function $\check{z}_3$, already discussed above, moderates this effect by encouraging orders to be served as early as possible, even beyond $T_{\ell+1}$. Objective $\check{z}_3$ has a similar structure to that of objective $\check{z}_2$, but assigns to each served order a revenue that decreases linearly depending on the period (re-optimization cycle) the order is served (as in Figure 6.1).

In particular, consider a DVRPMB instance with L re-optimization cycles of known duration, and the solution of the re-optimization problem during cycle ℓ. Let $\xi_p^{\gamma_i}$ denote the profit obtained by serving an order $i \in N$ during time interval $(T_\gamma, T_{\gamma+1}]$, $\gamma \in \{1,2,\ldots,L-\ell\}$. Then profit $\xi_p^{\gamma_i}$ is provided by the following Equation:

$$\xi_p^{\gamma_i} = \xi_p - \left(\frac{\gamma_i - 1}{L - \ell}\right)\xi_p \qquad \forall i \in N \tag{6.10}$$

Finally note that objectives $\check{z}_2$ and $\check{z}_3$ reduce to objective $\check{z}_1$ if $\xi_p = 0$ and $\xi_c = 1$.

6.2.3 Discussion regarding the terms of the Objective Function

Here we discuss some fundamental aspects of the objective functions introduced in Section 6.2.1 and 6.2.2. The first statement compares the routing costs obtained when using objectives $\check{z}_1$ or $\check{z}_2$.

Statement 1

Given that all customer orders are known, the routing cost $O(\check{z}_1)$ of the solution to the re-optimization problem obtained under objective $\check{z}_1$, is always lower than or equal to the routing cost $O(\check{z}_2)$ obtained under objective $\check{z}_2$.

This statement points out the obvious. Recall that both objectives will yield equal number of DO to be served. The optimal way of serving all orders (static and dynamic) in terms of routing cost will be the one obtained under $\check{z}_1$, and therefore, $O(\check{z}_1) \leq O(\check{z}_2)$. The same statement holds of course for the routing cost under objective $\check{z}_3$, i.e. $O(\check{z}_1) \leq O(\check{z}_3)$.

Statement 2

The optimal solution that maximizes customer service and minimizes routing costs (objective $\check{z}_1$) cannot guarantee superior solutions in terms of dynamic orders served for the overall routing problem (multiple re-optimization cycles).

We illustrate the above statement with two examples that address the overall routing problem (entire working period). In the first (Example 1) the solution obtained under objective $\check{z}_2$ for the entire dynamic routing problem includes a greater number of DO w.r.t. the solution obtained under $\check{z}_1$. This is typically the case as illustrated experimentally in Section 6.4.4. In the second example (Example 2), the reverse is observed; that is, more DO are served in the entire routing problem under $\check{z}_1$ compared to the solution of the problem under $\check{z}_2$. However, the latter is not a typical outcome but rather an exception.

<u>Example 1</u>

Consider the instance of Figure 6.3 with the depot located at the origin O. Customers are located at the endpoints of five (5) neighboring vertices of a regular octagon with center at point O and $R = 1$. The side of the octagon is $a = 2 * R * \sin\left(\frac{\pi/4}{2}\right) = 0.77$. Two vehicles are available, and the maximum duration of each route (T_{max}) (by assuming the vehicle speed to be 1) equals to 4.4.

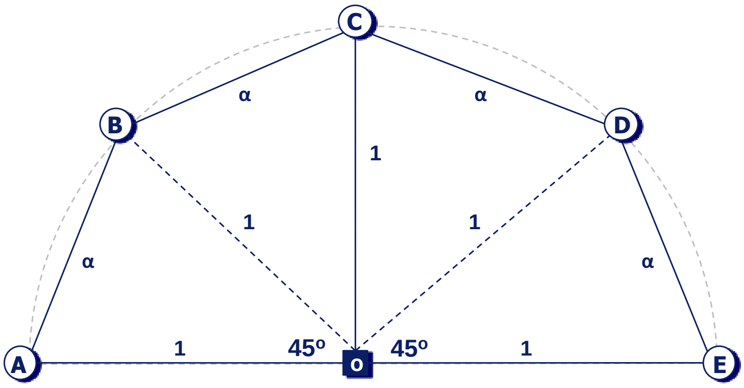


Figure 6.3. Customer topology for Example 1 of Statement 2

Assume that during $\ell = 0$ ($T_\ell = 0$), only customers A, B and C are known and a vehicle (K_1) has been assigned to serve the route $[O - A - B - C - O]$; the length of this route is equal to 3.54. The second vehicle (K_2) is available at the depot to be used as necessary. During the course of implementing this plan, DO D and E are received at times $h_D = 1.25$ and $h_E = 2.5$, respectively. Re-optimization takes place at fixed intervals, i.e. every 1.25 units of time.

For this scenario, we study the effect of objectives $\check{z}_1$ and $\check{z}_2$ considering two (2) re-optimization cycles. In Figure 6.4 we present for each objective and re-optimization cycle ($\ell > 0$) the state of the system prior to re-optimization ("*Before*") and the state after re-optimization ("*After*").

During $\ell = 1$ (at time $T_1 = 1.25$), vehicle K_1 is *en route* to customer B (under both objectives), while customer D is to be incorporated in the current plan. Solution under objective $\check{z}_1$ incorporates customer D with the best possible routing cost, i.e. immediately after customer C, yielding an overall routing cost (or, working time) of $O(\check{z}_1) = 4.31$. On the other hand, objective $\check{z}_2$ seeks to serve as many orders as possible within the current and the next re-optimization cycle and not just the next one, i.e. during time interval $[1.25,2.5]$. The optimal solution under such an objective is customer D to be served by vehicle K_2 located at the depot (K_2 arrives at customer D at time 2.5). This solution provides an overall routing cost $O(\check{z}_2) = O(K_1) + O(K_2) = 3.54 + 2 = 5.54$.

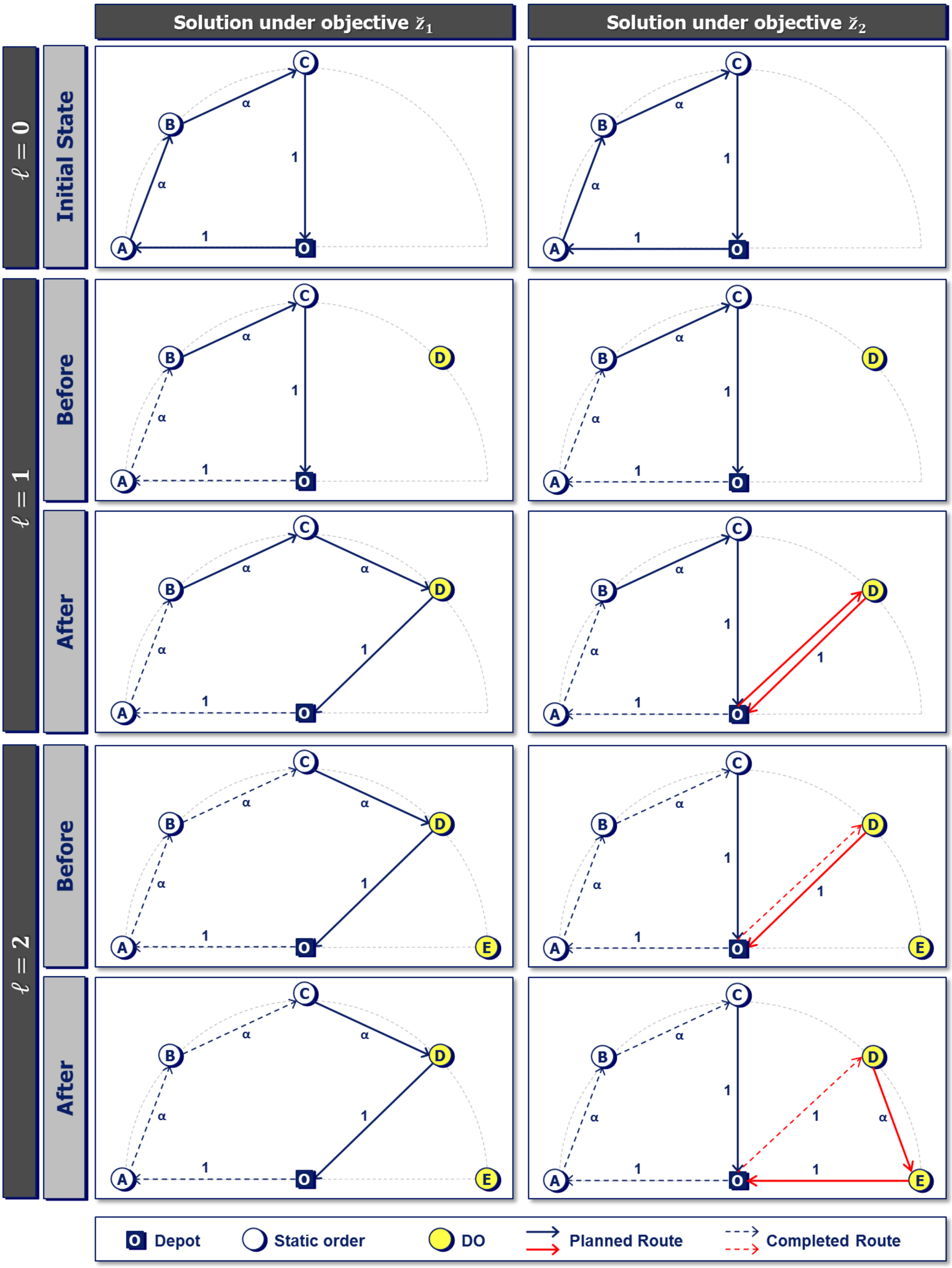


Figure 6.4. Example 1 for Statement 2 (blue color corresponds to vehicle $\boldsymbol{K_1}$ and red to $\boldsymbol{K_2}$)

The DO for customer E is received prior to $\ell = 2$ (at time $T_2 = 2.5$),. Vehicle K_1 is located at customer C under both objectives, while vehicle K_2 is located at the depot for the solution under objective $\check{z}_1$ and at customer D for the solution under objective $\check{z}_2$. For $\check{z}_1$, customer E can be served either by K_1 with total time of 5.08 or by K_2 with total time of 4.5 (since K_2 can begin service at $T_\ell = 2.5$); both of these options exceed the available working horizon (T_{max}), which means that customer E is not served under objective $\check{z}_1$. However, customer E can be served under objective $\check{z}_2$ by vehicle K_2 with total working time of 4.22, which is within the available working horizon. Thus, objective $\check{z}_2$ leads to a superior solution in terms of number of customers served. This example clearly illustrates that the objective that maximizes the number of served DO and minimizes routing costs, may lead to inappropriate commitment of resources w.r.t. the future state emerging after new DO are received.

<u>Example 2</u>

Consider the example of Figure 6.5. The depot (0), customers A, B, C and their coordinates along with important distances are shown in Figure 6.5, with $a = 0.9$ and $b = c = 0.56$.

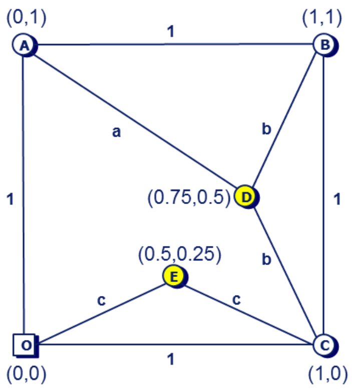


Figure 6.5. Network representation for Example 2 of Statement 2

As before, during $\ell = 0$ ($T_\ell = 0$), a vehicle has been assigned to perform route $[O - A - B - C - O]$, serving all static customers with cost equal to 4. During the course of implementing this plan, DO D and E are received at times $h_D = 1$ and $h_E = (1 + a + b) = 2.46$, respectively. Assume that two re-optimization cycles take place at fixed time instances (known in advance), i.e. $T_1 = 1$ and $T_2 = 1 + a + b = 2.46$. The total planning horizon (T_{max}) is equal to 4.5.

As before, Figure 6.6 illustrates the two system states per objective and re-optimization cycle for $\ell > 0$: prior to re-optimization ("*Before*") and after re-optimization ("*After*").

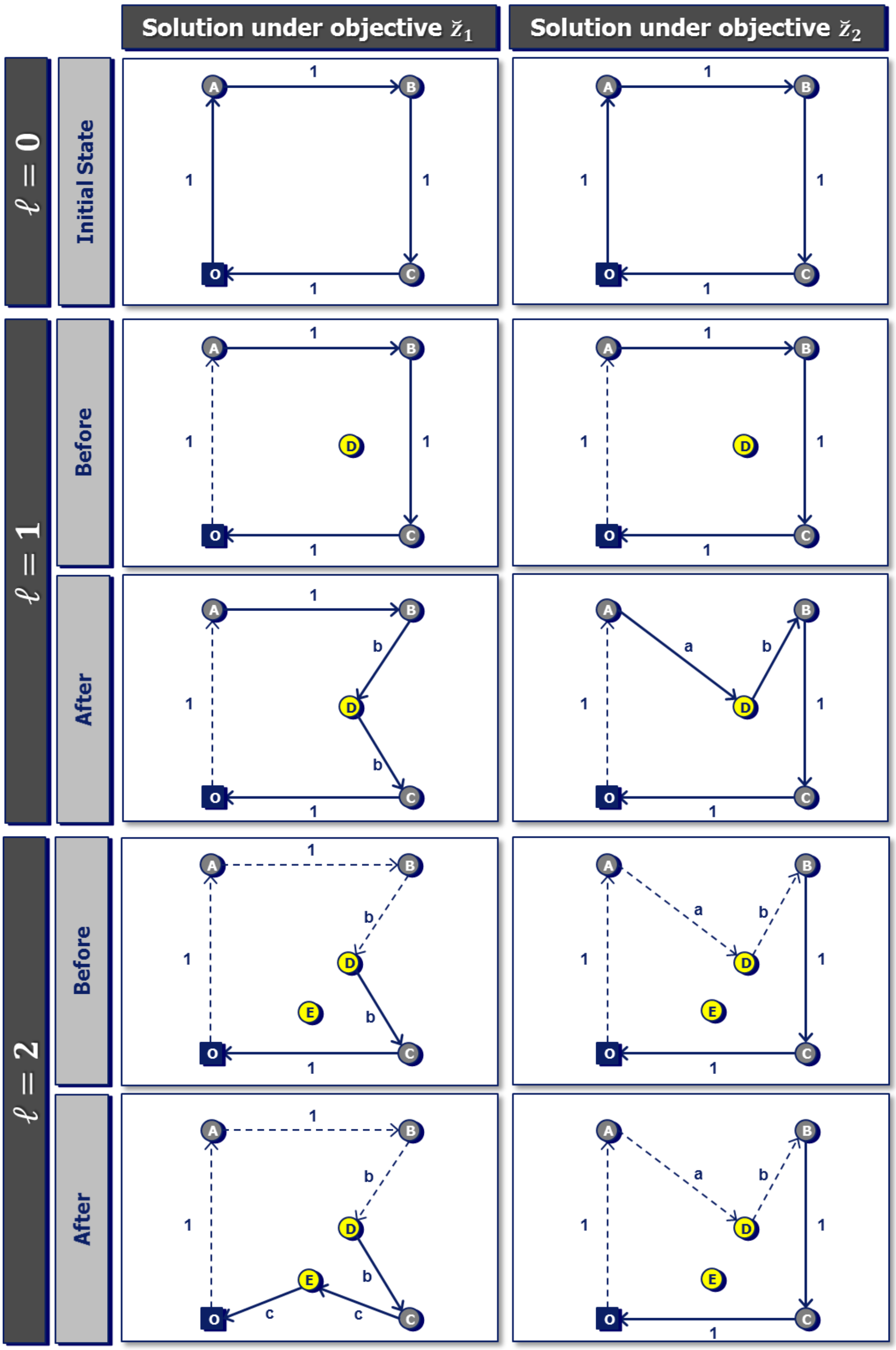


Figure 6.6. Solution provided under both objectives for Example 2 of Statement 2

At $\ell = 1$ (time $T_1 = 1$), the vehicle is located at customer A. The solution under objective $\check{z}_1$ incorporates customer D with the best possible routing cost, i.e. immediately after customer B, yielding an overall routing cost of $O(\check{z}_1) = 3 + 2b = 4.12$. Objective $\check{z}_2$ seeks to serve as many orders as possible within interval $[1,1 + a + b\ )$; this yields a solution in which customer D is visited after customer A; the overall routing cost is $O(\check{z}_2) = 3 + a + b = 4.46$.

At $\ell = 2$ ($T_2 = 1 + a + b$), the DO for customer E has been received and is to be served by the available vehicles. For the solution under objective $\check{z}_1$, the vehicle is *en route* to customer D, and customers A and B have already been served. Customer E can be incorporated in the current plan after customer C (i.e. route $D - C - E - 0$) with overall routing cost $O(\check{z}_1) = 2 + 2b + 2c = 4.24$. For the solution obtained under objective $\check{z}_2$, the vehicle is currently located at customer B and customers A and D have been served. The optimal inclusion of customer E (i.e. route $B - C - E - 0$) yields a cost of $O(\check{z}_1) = 2 + a + b + 2c = 4.58$, which exceeds the total available working horizon. Thus, customer E will remain unserved in this case, indicating that there might be cases where $\check{z}_1$ offers higher customer service compared to $\check{z}_2$. However, according to the experimentation of Section 6.4.4, this is not a typical outcome, but an exception.

In Section 6.4.4 we investigate experimentally under which conditions each of the investigated objectives is favored or not.

6.3 Modifications in the B&P algorithm to deal with limited resources

In this Section we present the necessary modifications of the B&P algorithm that solves the re-optimization problem of Chapter 4 (both exact and heuristic) in order to address the case of limited resources.

6.3.1 Modifying the Set-Partitioning formulation

In order to formulate the *m*-DVRPMB as a set-partitioning problem, the following should be addressed: a) incorporate the objective function described in Section 6.2 above, b) ensure that each delivery order is served (once), whereas each DO is served at most once, and c) limit the number of fleet resources.

Using the notation presented in Chapter 4, let A_r denote the set of orders served by route $r \in \Omega$, where Ω refers to the set of all feasible routes. Let e_{ij} be a binary coefficient that takes the value 1 if $i \in A_r$, and let ω_{ir} be a binary coefficient that indicates whether an order $i \in N$ is served during the time interval $[T_\ell, T_{\ell+1}]$ by route r or not. Also, let c_r denote the routing cost of route $r \in \Omega$. Finally, let ξ_u^i reflect the revenue if order $i \in F$ is served. For simplicity and without loss of generality, let $\xi_u^i = \xi_u, \forall i \in F$ and $\xi_u^i = 0, \forall i \in C$. Thus, the total cost $\tilde{c}_r$ of route $r \in \Omega$ for objective $\check{z}_2$ is given from:

$$\tilde{c}_r = \xi_c * c_r - \sum_{i \in A_r} (\xi_u^i + \xi_p * \omega_{ir}) \qquad \forall r \in \Omega \qquad (6.11)$$

For objective $\check{z}_3$, the total cost $\tilde{c}_r$ is:

$$\tilde{c}_r = \xi_c * c_r - \sum_{i \in A_r} (\xi_u^i + \xi_p - \left(\frac{\gamma_{ir} - 1}{L} \times \xi_p\right)) \qquad \forall r \in \Omega \qquad (6.12)$$

where γ_{ir} denotes the re-optimization cycle (time interval) in which customer $i \in N$ is served in route $r \in \Omega$ with $\gamma_{ir} \in \{1,2,\dots,L-\ell\}$ (where ℓ represents the current re-optimization cycle).

It should be noted that the total cost $\tilde{c}_r$ of route $r \in \Omega$ for objective $\check{z}_1$ is given from Eq. (6.11) when $\xi_p = 0$ and $\xi_c = 1$; i.e. $\tilde{c}_r = c_r - \sum_{i \in A_r} \xi_u^i$.

Recall from Chapter 4 (Section 4.2) that Ω (set of all feasible routes-columns) in our formulation comprises two subsets, i.e. $\Omega = (\bigcup_{k \in K} \Omega_k) \cup \Omega_p$, where columns Ω_k correspond to vehicles K *en route* and columns Ω_p to vehicles K_d located at depot. Consequently, the set partitioning problem for the Master Problem of *m*-DVRPMB may be formulated as follows:

$$(\boldsymbol{LP-SPP}) \qquad Minimize \qquad \sum_{r \in \Omega'} \tilde{c}_r y_r \qquad (6.13)$$

$$subject\ to: \qquad \sum_{r \in \Omega'} e_{ir} y_r = 1 \qquad \forall i \in C \qquad (6.14)$$

$$\sum_{r \in \Omega'} e_{ir} y_r \leq 1 \qquad \forall i \in F \qquad (6.15)$$

$$\sum_{r \in \Omega_p} y_r \leq |K_d| \qquad (6.16)$$

$$y_r = \{0,1\} \qquad \forall r \in \Omega \qquad (6.17)$$

Objective function (6.13) minimizes the total net cost of the selected routes. Constraint (6.14) ensures that each static order is visited by exactly one vehicle, while Constraints (6.15) state that each DO can be visited at most once. Finally, Constraint (6.16) limits the number of vehicles.

6.3.2 The Subproblem and its solution procedure

As already described in Chapter 4 (Sections 4.2 and 4.3), initially we construct a set of columns Ω' based on the solution at re-optimization timestamp T_ℓ and solve a restricted version of the

Master Problem (RMP). In order to check if this solution is globally optimal for the MP, we need to calculate the reduced costs $\hat{c}_r$ of each non-basic route $r \in \Omega \backslash \Omega'$. The reduced cost $\hat{c}_r$ of a route $r \in \Omega \backslash \Omega'$ for the *m*-DVRPMB is given by:

$$\hat{c}_r = \tilde{c}_r - \sum_{i \in C \cup F} e_{ir}\pi_i - \bar{\pi}_k \qquad \forall r \in \Omega \backslash \Omega', \forall k \in V \tag{6.18}$$

where π_i ($i \in C \cup F$) are the shadow (dual) prices related to customer Constraints (6.14) - (6.15), and $\bar{\pi}_k$ ($k \in K_d$) are the shadow (dual) prices related to resource Constraints (6.16).

Working along the same lines as in Chapter 4, the next step is to generate routes $\grave{r} \in \{\Omega \backslash \Omega'\}$ that have not yet been included in the current RMP, along with their reduced costs $\hat{c}_{\grave{r}}$. To do so, we solve the $|K| + 1$ sub-problems and for each sub-problem $k = 1,2, \ldots, |K| + 1$ the route $\grave{r}^*$ with the minimum reduced cost is derived based on Eq. (6.19) (see also Section 4.3);

$$\hat{c}^k_{\grave{r}^*} = \min_{\grave{r}} \left(\sum_{i \in N} e_{i\grave{r}} c'_{ij} - \bar{\pi}_k \right) \tag{6.19}$$

where c'_{ij} is the modified cost associated with arc $(i, j) \in A$. Specifically, for *m*-DVRPMB, the modified costs are given by:

$$c'_{ij} = \xi_c * c_{ij} - \xi^i_u - \pi_i \tag{6.20}$$

Recall from Chapter 4 that the scope of each sub-problem is to define the values of coefficients $e_{i\grave{r}}$ that minimize the related reduced cost. In order to formulate the ESPPRCTW sub-problem for *m*-DVRPMB, we modify appropriately the objective function (4.14) with Eq. (6.21) below:

$$min \sum_{(i,j) \in A} c'_{ij} x_{ij} - \sum_{\gamma \in L'} \sum_{i \in N} \xi^\gamma_p \omega^\gamma_i \tag{6.21}$$

where ω^γ_i denotes a decision variable that is equal to 1 if order $i \in N$ is served during time interval $[T_\gamma, T_{\gamma+1}]$, $\gamma \in L'$, where $L' = \{\ell', \ell' + 1, \ldots, L\}$ (ℓ' denoting the current re-optimization cycle) and 0 otherwise. Profit ξ^γ_p is calculated according to Section 6.2.2 (depending on the objective). In addition to Constraints (4.15)-(4.22) of the original formulation of Chapter 4, we also introduce Constraints (6.22)-(6.24) below. It should be noted that variables ω^γ_i do not participate in the solution under objective $\check{z}_1$, since for this case $\xi^\gamma_p = 0$.

$$w_i \in [T_\ell, T_{\ell+1}) \Rightarrow \omega_{i\ell} = 1 \qquad \forall i \in N, \ell \in \tilde{L} \tag{6.22}$$

$$\sum_{\ell \in \tilde{L}} \omega_{i\ell} \leq 1 \qquad \forall i \in N \tag{6.23}$$

$$\omega_{i\ell} \in \{0,1\} \qquad \forall i \in N, \ell \in \tilde{L} \tag{6.24}$$

Constraint (6.22) ensures that variable $\omega_{i\ell}$ will be equal to 1 if customer $i \in N$ is served within re-optimization cycle $[T_\ell, T_{\ell+1})$, $\ell \in \tilde{L}$, where $\tilde{L} = \{\ell'\}$ when objective $\check{z}_2$ is employed and $\tilde{L} = \{\ell', \ell'+1, \ldots, L-1\}$ when objective $\check{z}_3$ is used. Finally, Constraints (6.23) ensure that each order will be served only once during all re-optimization cycles and Constraints (6.24) force variables $\omega_{i\ell}$ to assume binary values. Constraints (6.22) can be linearized using Ineq. (6.25) below (where Z is a large positive number), ensuring that variables $\omega_{i\ell}$ will be equal to zero (i.e. will not be considered by the objective function) when an order $i \in N$ does not participate in the final solution (i.e. when $w_i = 0$).

$$Z(1-\omega_{i\ell}) + T_\ell \leq w_i \leq T_{\ell+1} - Z(1-\omega_{i\ell}), \qquad \forall i \in N, \ell \in \tilde{L} \tag{6.25}$$

Based on the above, the final model of the ESPPRCTW sub-problem for the *m*-DVRPMB comprises objective function (6.21) and the set of Constraints (4.15)-(4.22), (6.23)-(6.24) and (6.25).

Solution of the pricing sub-problem in *m*-DVRPMB

We solve the pricing sub-problems with the *label correcting algorithm* (Feillet *et al.*, 2004; 2005) described in Chapter 4 (Section 4.4). The application of the label correcting algorithm is straightforward when objective $\check{z}_1$ is employed, based on the aforementioned modifications. For objectives $\check{z}_2$ and $\check{z}_3$ we calculate profit ξ_p afresh during the extension functions within the label correcting algorithm (see Chapter 4, Eqs. (4.24) – (4.26)). In particular, Eq. (4.24) (in Section 4.4), which describes the extension function for the accumulated reduced cost of label $\Lambda_{\delta i}$ when extending to node j (resulting to new partial path δ'), is re-written as follows:

$$\tilde{c}_{\delta' j} = \begin{cases} \tilde{c}_{\delta' i} + c'_{ij} - \xi_p^{\gamma_j}, & when\ t_{\delta' j} < T' \\ \quad \tilde{c}_{\delta' i} + c'_{ij} & otherwise \end{cases} \tag{6.26}$$

where $\xi_p^{\gamma_j} = \xi_p$ and $T' = T_{\ell'+1}$ when objective $\check{z}_2$ is employed, and $\xi_p^{\gamma_j} = (\xi_p - (\frac{\gamma_j - 1}{L'} \times \xi_p)$ and $T' = T_{max}$ when objective $\check{z}_3$ is used.

6.4 Computational experiments

The experimental analysis is described in four Sections: Section 6.4.1 presents the test problems employed and the metric used for evaluating the results. Section 6.4.2 assesses the performance of the re-optimization B&P heuristic (of Chapter 4) w.r.t. its exact counterpart in solving the re-optimization problem under limited resources. Section 6.4.3 investigates the performance of those re-optimization strategies for which re-optimization is triggered based on the number of received DO; in this part of the study we employ only objective $\check{z}_1$, since objectives $\check{z}_2$ and $\check{z}_3$ may only be used under known re-optimization times. Finally, in Section 6.4.4 we compare the performance of objective functions $\check{z}_2$ and $\check{z}_3$ w.r.t. $\check{z}_1$ considering the entire dynamic problem.

All experiments have been implemented in Matlab® 7.14.0 (R2012a) using an Intel Core i7 PC System with processor speed 2.8 GHz and 4.00 GB of RAM running Windows 7®.

6.4.1 Experimental setup

6.4.1.1 Test instances

For the experimental study we have employed the R1 and C1 datasets of Solomon (12 and 9 instances, respectively), as described in Chapter 5 (see Section 5.4.1.2). We have also employed instances R100 and C100 that have no TW, but use the same customer coordinates as the R1 and C1 datasets. Table 6.1 summarizes the instances employed.

Table 6.1. Test instances

Distribution	Time-window	# Instances	Instances
Uniform	YES	12	R101, R102 ,R103, R104, R105, R106, R107, R108, R109, R110, R111, R112
Clustered	YES	9	C101, C102, C103, C104, C105, C106, C107, C108, C109
Uniform	NO	1	R100
Clustered	NO	1	C100

We also investigate how customer service is affected by *fleet availability*, i.e. by the number of extra vehicles stationed at the depot to serve DO. To do so, for each one of the 23 instances, we examined three (3) cases of 0, 2 and 4 vehicles available at the depot (denoted as V-0, V-2, V-4, respectively). Figure 6.7 illustrates the average number of DO (as a percent of total) served per dataset (R1 and C1) for each value of vehicle availability at the depot, along with the minimum and maximum values. The percentages illustrated are the averages over all instances (and test problems) by assuming that all DO are known in advance (see Section 6.4.2).

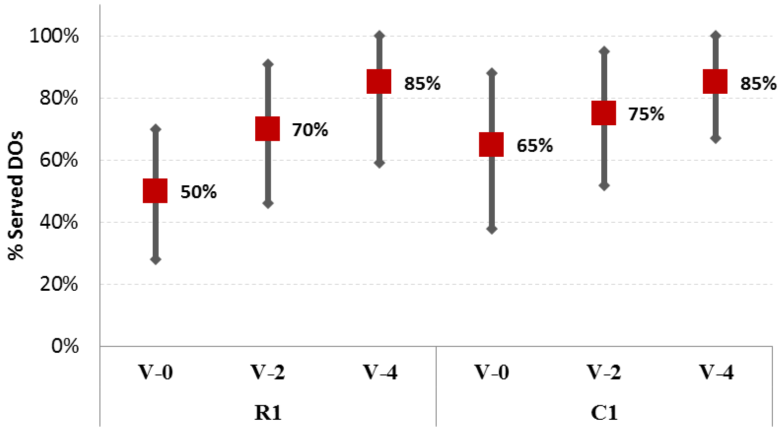


Figure 6.7. Average percentage of DO served vs. the available vehicles at depot per dataset

Thus, in total we constructed 69 different cases (3 x 23). For each case we assumed *moderate dod* (50% DO) and generated 10 different problems (different selection of offline requests), resulting in a total of 690 test problems. The generation process of the test problems and the process of generating the initial solutions remains the same as the one described in Chapter 5 (Section 5.4.1.2).

6.4.1.2 The metric used for comparison (value of information)

In this experimental investigation we report the performance of the proposed methods (for both the re-optimization problem and the entire dynamic one) based on the *value of information* (VoI), as described in Chapter 5 (Section 5.4.1.1). In this study, we enhance this metric in order to take into account both the number of DO served and the routing cost. Let $F_{\mathcal{F}}(\mathcal{H})$ denote the total number of DO served in the final solution of dynamic problem $\mathcal{H}$ under objective $\mathcal{F}$ and $\bar{\bar{C}}_{\mathcal{F}}(\mathcal{H})$ the corresponding total routing cost. Then, the value $z_{\mathcal{F}}$ of problem $\mathcal{H}$ when solved under $\mathcal{F}$ can be calculated as:

$$z_{\mathcal{F}}(\mathcal{H}) = -\xi_u * F_{\mathcal{F}}(\mathcal{H}) + \bar{\bar{C}}_{\mathcal{F}}(\mathcal{H}) \tag{6.27}$$

Since the objective function results in negative values, the VoI used in this Chapter is given by the following formula:

$$V_{\mathcal{F}}(\mathcal{H}) = \frac{z_{\mathcal{F}}(\mathcal{H}) - z_{\mathcal{F}}(\mathcal{H}_s)}{abs(z_{\mathcal{F}}(\mathcal{H}_s))} \times 100 \tag{6.28}$$

where $z_{\mathcal{F}}(\mathcal{H}_s)$ denotes the value of the metric for the related static problem $\mathcal{H}_s$ (in which all DO are known prior to the dispatching of the vehicles; i.e. at time $t = 0$). It should be also noted that for ξ_u was set to 1000 throughout the entire experimentation.

6.4.2 Assessment of the re-optimization B&P heuristic in m-DVRPMB

In this Section we assess the performance of the proposed heuristic B&P algorithm of Section 4.7 in solving the re-optimization problem with limited resources. To do so, we employ all test problems described in Section 6.4.1.1 by assuming that a) all static orders have been assigned to vehicles according to the methodology described in Section 5.4.1.2, and b) all DO are known before the vehicles are dispatched from the depot. Each test problem is then solved by both the exact (OPT) and the heuristic (HEUR) algorithm. Due to the fact that re-optimization takes place only once (since all DO are known in advance), both algorithms were executed under objective $\check{z}_1$.

Table 5.3 summarizes the results obtained per instance as an average over all test problems solved. For each instance, the Table reports the percentage deviation of the solution of HEUR from that of OPT *(%Dev*) in terms of VoI (as described in Section 6.4.1.2), and the computational times CT_{OPT} and CT_{HEUR} (in sec). The bottom Section of the Table reports the average performance indicators for the R1 and C1 datasets.

Table 6.2. Performance of heuristic B&P algorithm

Instance	%Dev	CT_{OPT}	CT_{HEUR}	Instance	%Dev	CT_{OPT}	CT_{HEUR}
R101	0.3%	9.6	15.2	**C101**	0.2%	10.1	21.2
R102	0.7%	40.9	16.6	**C102**	1.7%	98.4	25.3
R103	2.0%	1,255.2	21.2	**C103**	2.5%	459.2	57.2
R104	2.8%	1,496.2	70.5	**C104**	2.5%	1045.6	69.4
R105	1.1%	60.29	24.6	**C105**	1.5%	109.5	32.6
R106	1.8%	196.58	39.2	**C106**	1.9%	121.3	31.5
R107	2.3%	509.2	50.8	**C107**	2.1%	147.4	38.6
R108	2.6%	1,564.1	67.9	**C108**	2.9%	164.1	45.3
R109	1.6%	349.2	45.3	**C109**	2.3%	138.4	39.2
R110	1.7%	1,443.12	61.7				
R111	1.9%	1,202.83	65.9				
R112	2.4%	222.15	58.4				
Average R1	*1.8%*	*695.8*	*44.8*	***Average C1***	*2.0%*	*254.9*	*40.1*

Based on Table 5.3, HEUR seems to yield efficient solutions with an average deviation of 1.9% from the optimum over all instances, a performance similar to that of the HEUR in the unlimited fleet case (all DO served). Regarding the computational effort, HEUR seems to be highly efficient compared to its exact counterpart, as expected.

6.4.3 Performance of re-optimization strategies in *m*-DVRPMB when re-optimization depends on the number of DO received

In this Section, we investigate the performance of the re-optimization strategies for the limited fleet case. The main objective is to investigate the trends of the various strategies and compare them to those observed in the unlimited fleet case (Chapter 5, Section 5.4.3). In order to align our analysis in this Section to the one of Chapter 5, we employ re-optimization policies that depend on the number of DO received. Since the re-optimization cycles under such policies are not of known duration, we perform the current analysis only under $\check{z}_1$.

We employed all instances described in Section 6.4.1.1, and, similarly to Chapter 5, we used the SRR, NRR-1, NRR-2 and NRR-3 policies; each policy was tested under the FR and PR release tactics, resulting to a total of eight (8) strategies for each one of the 690 test problems (i.e. 5,520 problems in total). It is noted that the analysis of the experimental results uses appropriate averages. The detailed results of the strategies for all instances and for the different values of fleet availability have been included in Appendix B (Table B.1).

Figure 6.8 presents the performance (w.r.t. VoI) of each re-optimization strategy (policy-tactic combination) for each investigated dataset (R1 and C1), averaged over all test problems of the related dataset (and of course over all cases w.r.t. the number of vehicles available at the depot). From this Figure it is clear that the SRR-PR strategy provides the best average performance (minimum VoI) and the PR tactic outperforms FR (on the average) in all datasets. The performance difference between the two tactics decreases as the number of elapsed DO per re-optimization cycle increases (less number of re-optimization cycles). Furthermore, the PR tactic seems to be more efficient for shorter re-optimization cycles and the FR tactic seems to be less efficient when re-optimization is applied very frequently (SRR) or infrequently (NRR-3). The observed performance seems to agree with the behavior of the re-optimization strategies for the unlimited fleet case (see Section 5.4.3, Fig. 5.9), indicating that the performance of the strategies is not affected significantly by limiting the available resources.

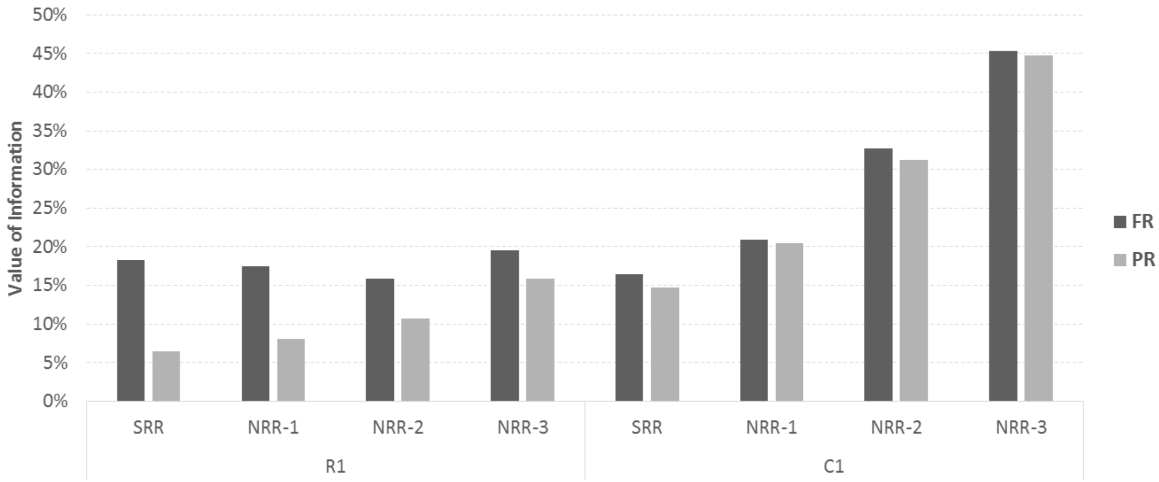


Figure 6.8. Average performance of re-optimization strategies for the different datasets

In terms of the interaction of the re-optimization strategies with fleet availability, Figure 6.9 and Figure 6.10 present the performance of each tactic w.r.t. the policies for the R1 and C1 datasets, respectively, and for different values of fleet availability (V-0, V-2 and V-4). Note that this is the average performance over all related instances (and test problems). The Figures illustrate similar patterns with the aforementioned analysis w.r.t. the performance of re-optimization strategies for all values of fleet availability, i.e. there is no significant interaction between fleet availability and the re-optimization strategies on the average.

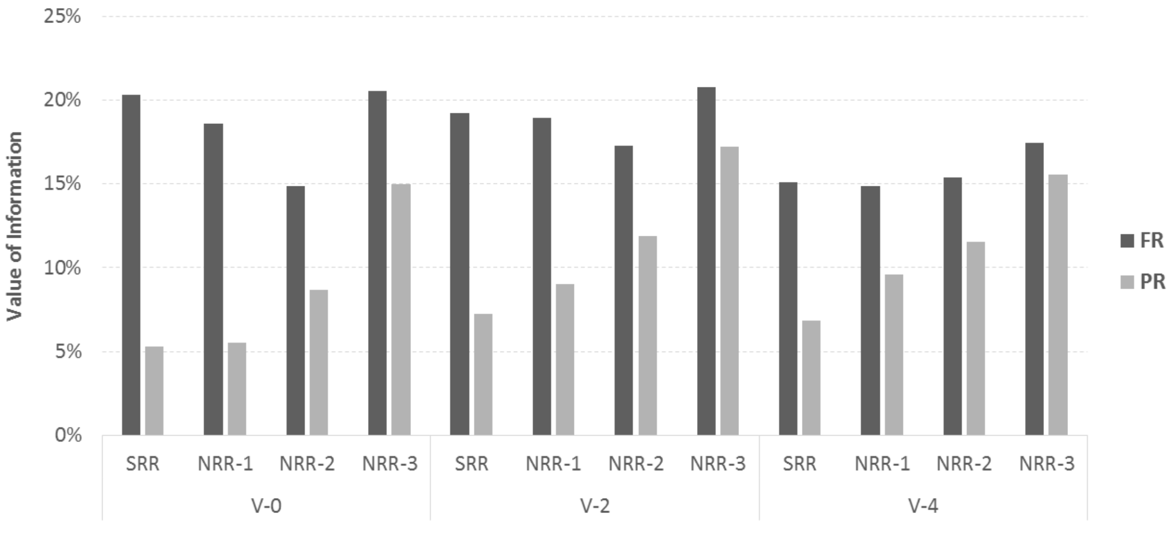


Figure 6.9. Performance of strategies for **R1** dataset for various values of fleet availability

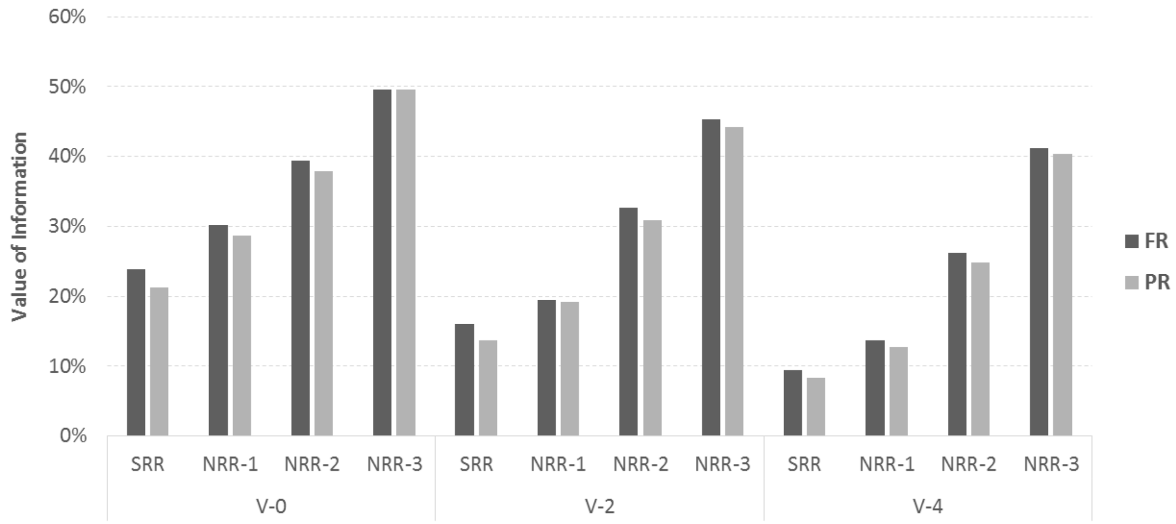


Figure 6.10. Performance of strategies for **C1** dataset for various values of fleet availability

6.4.4 Performance of proposed objective functions under re-optimization cycles of known duration

We assess the performance of the three proposed objectives discussed in Section 6.2, i.e. i) $\check{z}_1$, which provides a fixed profit for each DO served; ii) $\check{z}_2$, which provides an additional profit for each order served within the next re-optimization period (in lexicographical order), and iii) objective $\check{z}_3$, which assigns profit to all orders, with the profit decreasing linearly depending on the period the order is served (in lexicographical order as well).

The investigation of this Section includes all instances described in Section 6.4.1.1 for the R1 dataset (13 instances, including R100), under three values of fleet availability (V-0, V-2 and V-4), and using 10 different test problems per instance (i.e. 390 test problems in total).

Since objectives $\check{z}_2$ and $\check{z}_3$ may be used only if the re-optimization time instances (and intervals) are known in advance, we employed fixed-time re-optimization policies (FTR policies; see Chapter 5.2), which comprise cycles of equal duration. In particular, we investigated four values of re-optimization frequency, i.e. every 10, 20, 40 and 60 units of time w.r.t. T_{max} (which is equal to 230 units of time in the Solomon instances), hereafter designated as FTR-10, FTR-20, FTR-40 and FTR-60. Each policy was tested under the FR and PR release tactics, resulting to a total of eight (8) strategies for each one of the 390 test problems of R1 dataset (i.e. 3,120 problems in total). The analysis of the experimental results uses appropriate averages. The detailed results of the re-optimization strategies for all instances, objectives and the different values of fleet availability are included in Appendix B (Tables B.2 – B.4).

Figure 6.11 presents the performance (w.r.t. VoI described in Section 6.4.1.2) of each objective for each investigated instance of the R1 dataset (incl. R100), averaged over all test problems and re-optimization policies and tactics. According to the Figure, objectives $\check{z}_2$ and $\check{z}_3$ (that consider vehicle productivity) seem to provide more efficient solutions for cases with increasing TW width, compared to objective $\check{z}_1$; this improvement is more pronounced in cases with wide TW (R103, R104, R107, R108) or no time windows (R100).

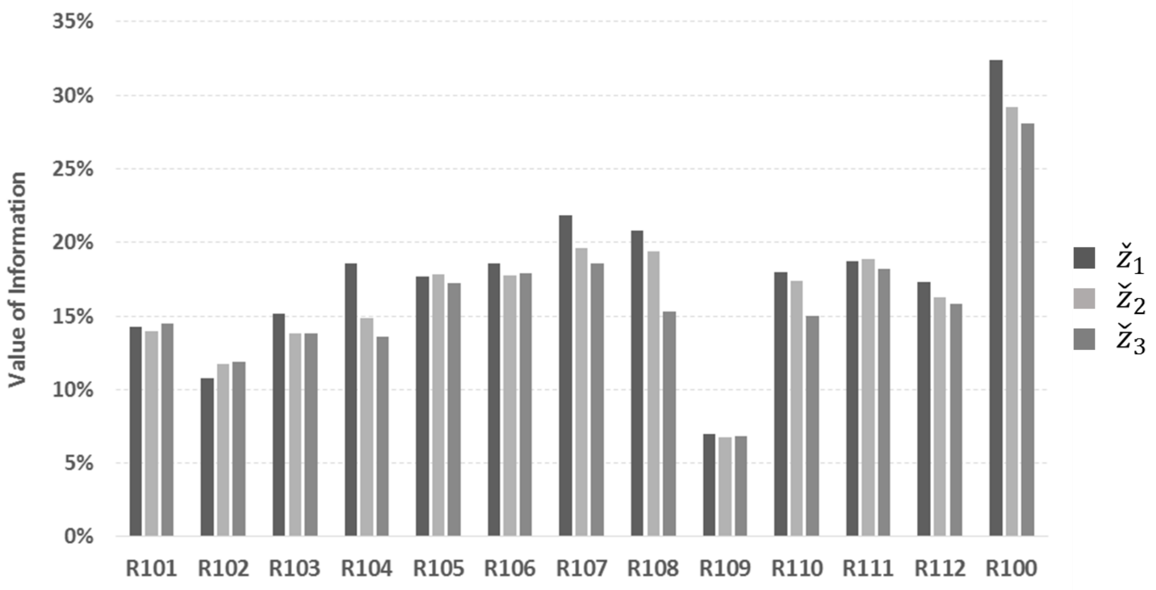


Figure 6.11. Overall average performance of objectives per investigated instance

This trend may be attributed to the fact that wide (or no) TWs allow for more customer combinations and, thus, more opportunities for customers to be served sooner (e.g. till the next re-optimization cycle). In addition, objectives $\check{z}_2$ and $\check{z}_3$ keep vehicles busy, delaying their return to the depot. This allows for increased opportunities when new orders are considered in subsequent cycles. On the other hand, $\check{z}_2$ and $\check{z}_3$ do not seem to favour the solution for cases with limited TW width (e.g. R101, R102, R105); the limited feasible timeslot for service of DO in those case decreases the possibilities of including future DO in the plan. This fact, in combination with the expected increase of routing costs under objectives $\check{z}_2$ and $\check{z}_3$ (based on Statement 1), can cause their performance to deteriorate compared to objective $\check{z}_1$.

Figure 6.12 illustrates the performance of the objectives with respect to re-optimization strategies (policy and tactic combination), averaged over all test instances and values of vehicle availability. Based on the Figure, objective $\check{z}_3$ seems to lead to more efficient solutions when re-optimization is applied more frequently (FTR-10 and FTR-20). This may be attributed to the fact that frequent re-optimization (e.g. FTR-10) leads to higher number of re-optimization cycles, thus allowing $\check{z}_3$ to allocate the DO to the appropriate period (but not forcing service of orders only on the upcoming re-optimization cycle as in $\check{z}_2$). On the other hand, objective $\check{z}_2$ performs slightly better in cases of longer re-optimization intervals.

Another interesting observation resulting from Figure 6.12 is that performance under objectives $\check{z}_2$ and $\check{z}_3$ improves in the FR tactic. This is expected, since using $\check{z}_1$ (and thus not accounting for vehicle productivity) under the FR tactic, may schedule the service of newly received DO way into the future in the expense of significant resources, e.g. time (especially for newly dispatched vehicles from depot). This is not the case with $\check{z}_2$ and $\check{z}_3$, which tend to use the vehicles *en route* as much as possible given the current information and the additional work to come.

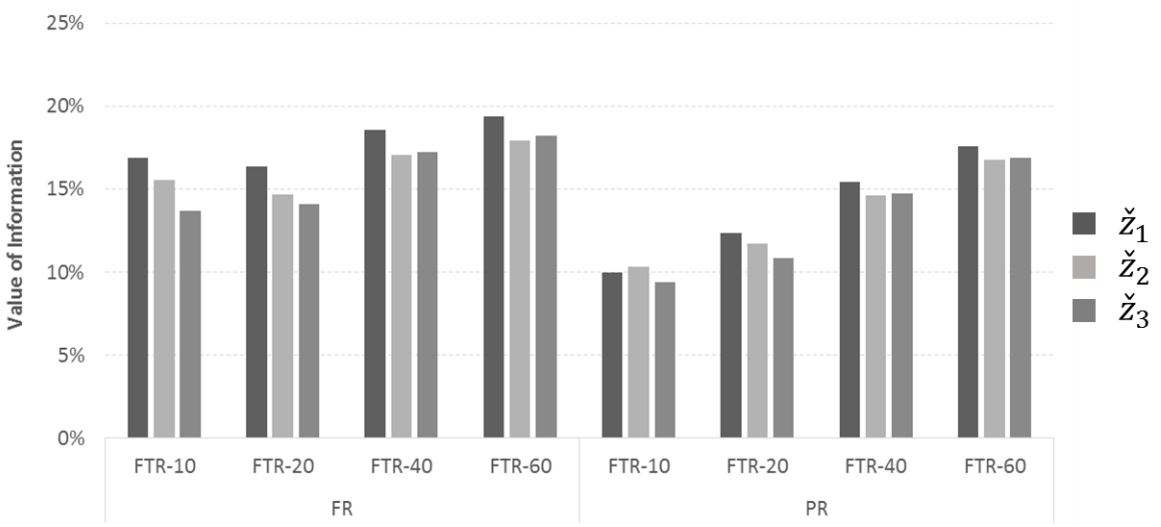


Figure 6.12. Average performance of objectives w.r.t. re-optimization policy and tactic

Finally, we investigate the interaction of different values of vehicle availability and different TW patterns (the latter characterized by the ratio of the average TW width of all customers w.r.t. T_{max}). To do so, we grouped all investigated instances in two categories, as shown in **Error! Not a valid bookmark self-reference.**.

Table 6.3. Classification of investigated instances in TW-pattern groups

Group	% of T_{max}	# Instances	Instances
TW-1	5% - 40%	7	R101, R102, R105, R106, R109, R110, R111
TW-2	>40%	6	R103, R104, R107, R108, R112, R100

Figure 6.13 presents the average performance of the objectives for the aforementioned TW-pattern groups and for the three values of vehicle availability. The results shown are averaged over all instances and re-optimization policies and tactics. The Figure illustrates that the higher

the number of available vehicles and the wider the TW, the better objectives $\check{z}_2$ and $\check{z}_3$ perform. This may be attributed to the tendency of objectives $\check{z}_2$ and $\check{z}_3$ to serve DO as early as possible, leading to additional flexibility of vehicles employed during future re-optimization cycles (when additional DO arrive). On the other hand, objective $\check{z}_1$ may schedule more DO to be served during future periods, limiting this flexibility.

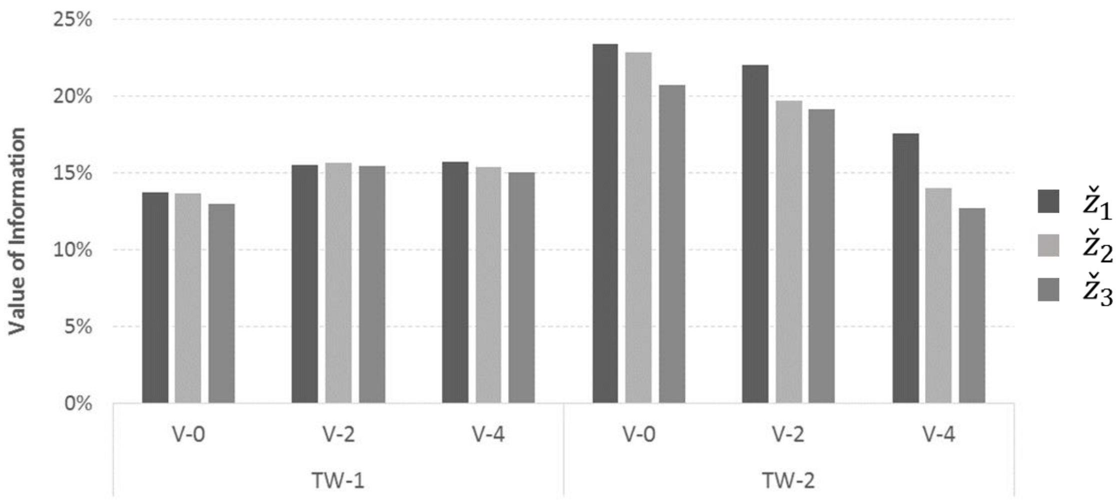


Figure 6.13. Average performance of objectives for different TW pattern groups and vehicle availability values

The above experimental analysis indicates that objectives which account for vehicle productivity are more appropriate for challenging cases (wide or no TW), or cases for which more than say 50-60% of DO may be served by the available fleet (more than 2 vehicles available at depot, according to Figure 6.7). In cases with narrow TW or limited fleet availability, accounting for vehicle productivity does not seem to help appreciably. Furthermore, for the preferred short re-optimization intervals (i.e. 5-15% of the available working horizon) using objective $\check{z}_3$ seems more efficient.

In order to put the aforementioned analysis into context, we present in Figure 6.14 the average performance of the objectives in terms of number of DO served for those parameters that favor objectives $\check{z}_2$ and $\check{z}_3$ (according to previous analysis), i.e.: a) V-4 regarding vehicle availability, and b) instances R104, R108 and R100 (wide or no TW). Results are reported w.r.t. re-optimization tactic and frequency; for the latter, we grouped FTR-10 and FTR-20 under category “Frequent” and FTR-40 and FTR-60 under category “Infrequent”. The Figure illustrates that, as discussed previously, objective $\check{z}_3$ is more appropriate for frequent re-optimization under FR tactic, offering up to about 15% more DO served; this is limited to about 4% when the PR tactic is employed. On the other hand, objective $\check{z}_3$ seems to perform best under infrequent re-optimization, offering up to 10% more DO served under the FR tactic and 4% under PR.

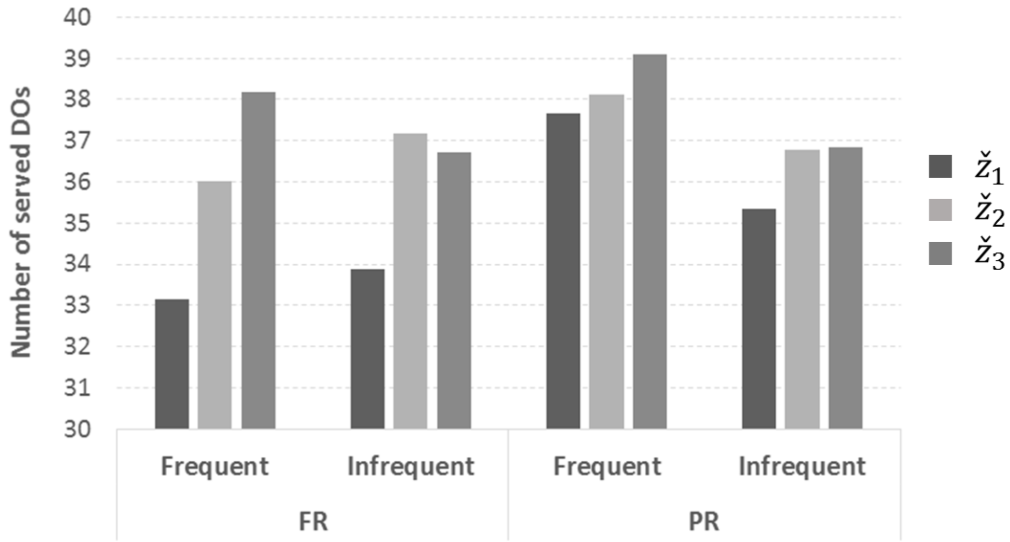


Figure 6.14. Average number of served DO per objective w.r.t. re-opt. frequency and tactic

Using objectives that account for vehicle productivity is recommended under operational settings with relatively high vehicle availability, wide TW and especially when the FR tactic is necessary.

6.5 Case study in a Courier environment

We have applied the proposed method for the DVRPMB with limited resources to a typical case in a next-day courier service provider. ELTA Courier is a part of the Greek Postal Service (ELTA) and has the third largest market share among all couriers operating in Greece. In addition to its own network, ELTA Courier uses the extended distribution network of the Greek Postal Service.

The case study was part of the project "*MADREL (Management of Dynamic Requests in Logistics)*" conducted in the DeOPSys Lab of the University of the Aegean. The project focused on the design, implementation and evaluation of an integrated system that supports planners and dispatchers to deliver enhanced courier operations. The MADREL system supports, in addition to the daily routing of all known orders, two significant activities: a) planning of mass deliveries over a multiple-day horizon (orders with flexible delivery dates within a pre-specified service level), and b) allocation of real-time dynamic orders (DO) that occur during service execution. The method for allocating mass deliveries solves a special variation of the multi-period VRP using a Branch-and-Price technique. For planning DO in real time, MADREL uses an efficient insertion heuristic. More information regarding the context of this project can be found in Ninikas *et al.* (2014).

In this Section we employ the real-life data used in testing the MADREL system and we apply our proposed B&P-based method for the allocation of DO. The resulting solutions are compared to a) those of the conventional approach followed by the dispatchers, and b) those obtained by the MADREL insertion-based heuristic mentioned above.

6.5.1 Current issues in courier distribution

Figure 6.15 presents a typical model of the courier supply chain. The distribution vehicles depart loaded early in the morning (e.g. 08:00) from a Local Service Point (LSP) to perform deliveries or pick-ups; typically, each delivery vehicle serves a certain geographical area. By the end of the shift, all vehicles return to the LSP having delivered their entire load (minus unserved returns), and carrying items that were picked-up. After processing all collected items (08:00 pm in the example of Figure 6.15), the LSP forwards them to the corresponding hub. In turn, the hubs, process and forward the items delivered by their LSPs, typically overnight, to the destination hubs. The latter, after processing, forward these items (e.g. by 6:00 am) to the LSPs, which are responsible for delivering them.

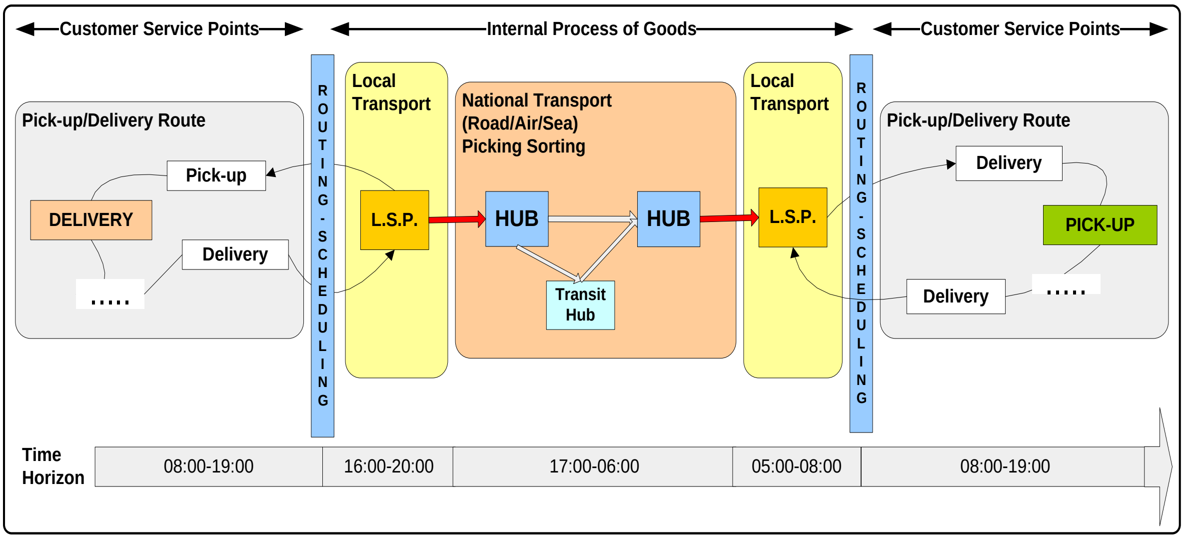


Figure 6.15. A typical model of courier service operations

For the delivery/pick-up operation, the LSP dispatcher typically knows in advance only a subset of the tasks. A number of requests for pick-ups of parcels/documents appear dynamically over time as the delivery plan is executed. As a result, vehicle routing includes a dynamic component, which makes it more challenging than typical (static) routing. In addition to daily pick-ups and deliveries, the LSPs also deal with mass deliveries. The promise dates of these deliveries have some flexibility within a pre-specified service level. For example, internet kits may be delivered to the clients within a week from the time of order, by providing a day's notice for the exact time of delivery.

Figure 6.16, overviews the typical planning and routing process followed by an LSP to deal with planned deliveries and pick-ups (*regular orders*), mass deliveries (*mass orders*), and requests for service during delivery execution (*dynamic orders*).

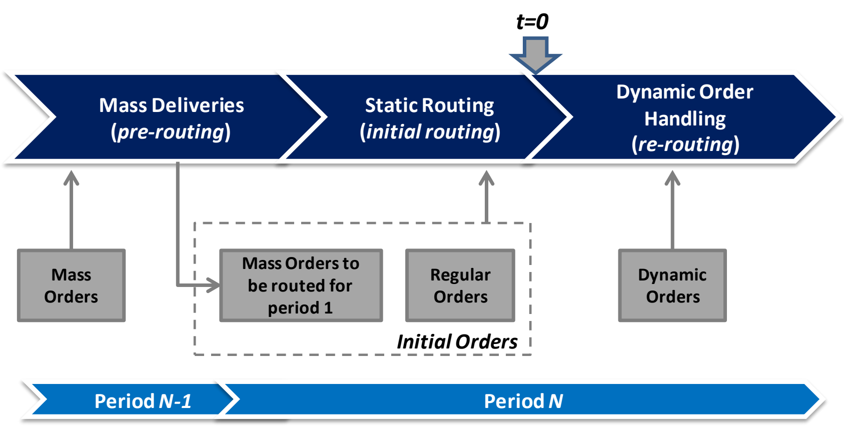


Figure 6.16. Typical routing process (the initial routing plan is generated at t=0)

Initially, the dispatcher allocates the mass orders to delivery days by taking into consideration the expiration day of each order. The second step involves routing of all known orders; that is, regular orders, as well as the mass orders allocated by the previous step to that particular day. The result of the first two steps is the synthesis of the initial routing plan. To incorporate the DO in the initial plan, the dispatcher re-routes certain vehicles. This is done dynamically throughout the shift.

In the following analysis, we deal only with the two latter steps of this integrated scheme for managing hybrid courier operations. For more information regarding a solution framework for handling the first step (allocation of mass deliveries), the reader can refer to Athanasopoulos and Minis (2011) and Ninikas *et al.* (2014).

6.5.2 Key data for the case study

The case study concerns a single LSP serving an urban area in Athens of average size (700 km^2). The LSP serves approximately 450 static (delivery) orders (SO) and 70 DO daily with a heterogeneous fleet of 5 vans and 8 scooters. The data collected comprise of a 3-day period (Tuesday to Thursday). All geographical data (x,y coordinates) of the customers involved (both static and dynamic) were provided from appropriate GPS devices installed on the vehicles of the corresponding LSP.

The pick-up and delivery demand from the customers correspond to either letters/small parcels that are typically handled by scooters, or medium to big parcels that fit only to vehicles (vans). Let N_S and N_V, denote customers of the former and the latter type, respectively; thus the total customer set is denoted as $N_T = N_S \cup N_V$. Note that N_V cannot be accessed by a scooter whereas N_S can be served by both vans and scooters. Table 6.4 presents key characteristics of the case study.

Table 6.4. Key indicators of the case study (number of customers and resources)

Day	Resources		Customers					
	Vans	Scooters	N_T		N_V		N_S	
			SO	DO	SO	DO	SO	DO
Day 1	5	8	477	68	150	26	327	42
Day 2	5	8	491	68	160	23	331	45
Day 3	5	6	370	66	146	30	224	36

Figure 6.17 represents the customers' locations over the LSP's service region on a digital map for the busiest day of the 3-day period (Day 2). The dispersion of the customers for the remaining two days has a similar pattern.

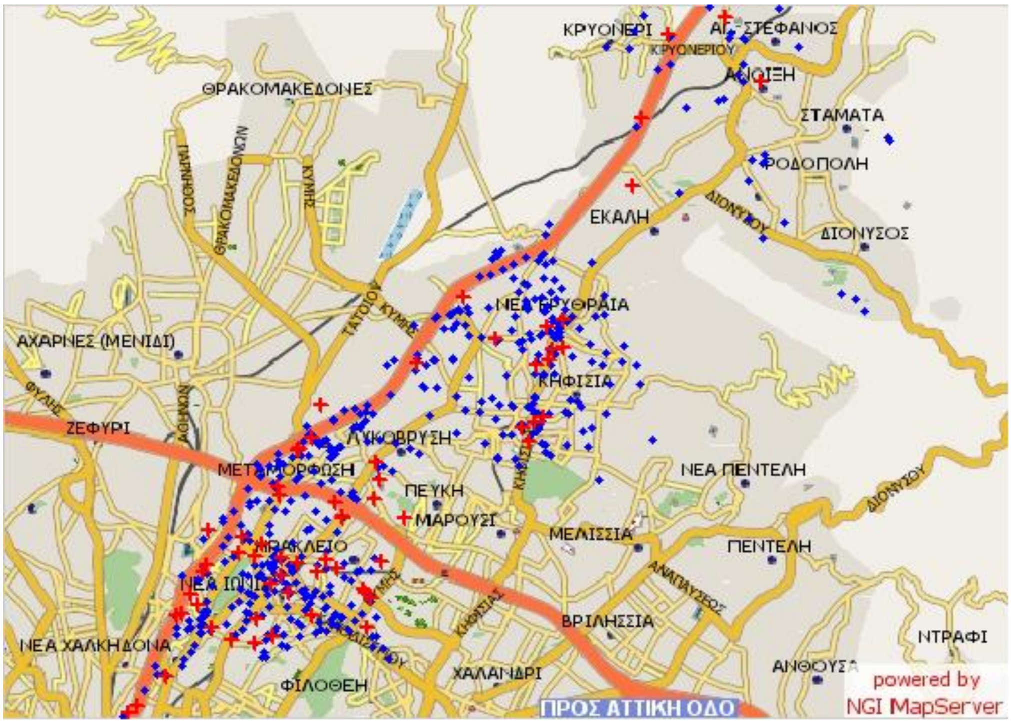


Figure 6.17. Representation of customer locations for the busiest day on a digital map (blue circles correspond to static customers and red crosses to DO)

A call-center is available and can receive calls (DO) requesting on-site pick-up from 08:00 till 18:00. DO typically request a time-window (TW), which normally ranges from 2 to 4 hours and starts within 30 to 60 minutes after the call has been received. Figure 6.18 and Figure 6.19 present the temporal distribution of DO as a function of the time of day (early morning and late afternoon hours are less busy) and arrival and TW pattern, respectively.

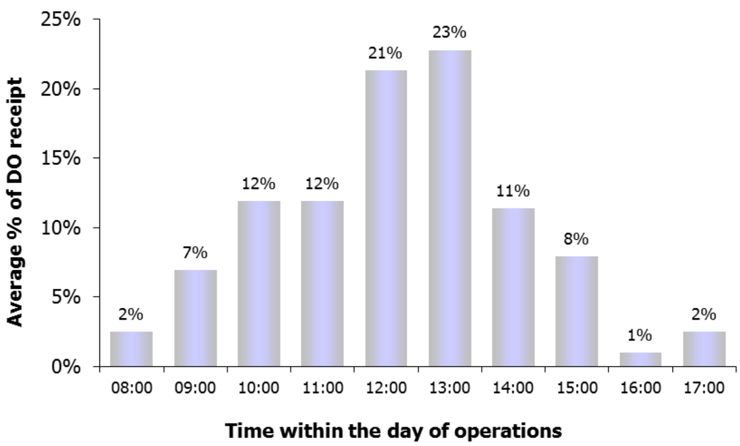


Figure 6.18. Distribution of DO received w.r.t. the time of day (all three days)

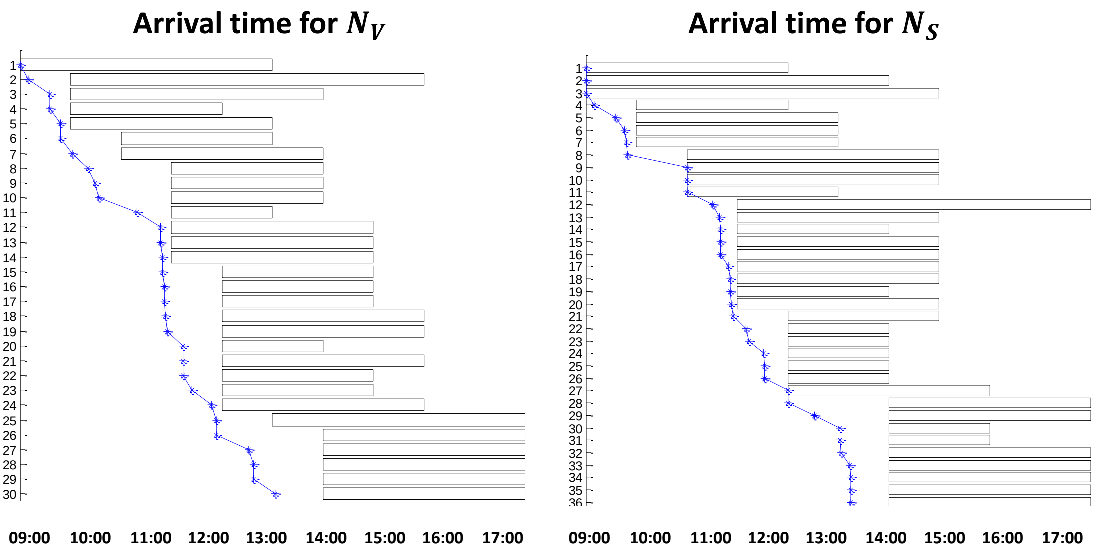


Figure 6.19. Arrival and TW patterns of DO for Day 3 (a blue star reflects the time each DO was received; the bars indicate the related TW)

Onsite service times at the customer location were recorded through GPS-based devices. It is worth mentioning that, on average, DO require almost 50% more on-site service time than static requests, mostly because they involve additional work from the driver.

Concerning the estimation of travel times, we analyzed the correlation between historical travel times and distances for a 2-month period (approximately 1000 locations per day) and used the results to estimate the travel time as a function of the Euclidean distance between two locations. We constructed the distance and time matrix based on these estimates.

Finally, Table 6.5 presents information regarding the average number of static (SO) and dynamic (DO) orders assigned to each van and scooter. Scooters are normally assigned on average 30% more orders than vans, since the former are able to travel faster in the congested city streets.

Table 6.5. Average number of customers served per vans and scooters

Order type	Average orders served per Van	Average orders served per Scooter
SO	30,4	39,9
DO	5,3	5,6
Total	*35,7*	*45,5*

6.5.3 The MADREL insertion heuristic

As mentioned above, for MADREL we developed an *insertion-based heuristic* in order to incorporate the available unserved DO in the current plan in a time-efficient manner. The complexity of the insertion heuristic is highly dependent on the number of DO and the number of available arcs (that a DO can be potentially inserted). Furthermore, since insertion is sequence dependent, an optimal insertion procedure is of factorial complexity $O(|F|!\sum_{i=1}^{N}(g + i - 1))$, where $|F|$ is the number of DO, and g is the number of available arcs in the planned routes. Thus, if the number of DO to be re-optimized is higher than say 8 or 9, then an exhaustive algorithm is computationally intractable. Thus, the current insertion-based heuristic considers the sequence dependency of neighbor DO only without evaluating all sequences.

The heuristic comprises three steps, as described below.

Step 1: Initialization.

The first step processes all available information up to re-optimization event T_ℓ; i.e., the remaining static orders which have yet to be served, the remaining capacity in both time and load of the vehicles *en route*, and the dynamic orders (DO). Concerning the DO, we consider two cases (as in Chapter 5.2); for re-optimization under *FR tactic*, we consider only the DO arrived during the interval $[T_{\ell-1}, T_\ell]$. For re-optimization under the *PR* tactic, we also consider DO that have arrived in $[T_0, T_{\ell-1}]$ but not served yet.

Step 2: Clustering of DRs

As discussed previously, it is assumed that the order of inserting DO in the routes is significant only when DO compete for the same arcs. Based on this assumption, the algorithm decomposes the entire set of DO to smaller subsets $l_n, n = 1,2, \dots, \Lambda$, each containing competing DO. Due to the complexity considerations discussed above, the number of DO per subset is kept low, $|l_n| \leq 6$. The clustering of DO to competing sets is performed as follows:

Step 2.1. For each DO, $i \in F$, the most preferable, feasible insertion arcs are determined and stored in set S_i. Feasibility refers to respecting order time windows, vehicle capacities and shift duration (T_{max}). The maximum number of feasible arcs stored in each S_i has been set to 10 (i.e. S_i contains up to ten of the most favorable and feasible arcs), which has been determined through experiments to be a fairly adequate number regardless the number of customer orders involved.

Step 2.2. The arc-set S_i of each DO $i \in F$, is compared with each set $S_j, j \in F, j \neq i$, to determine whether DO i and j compete for one or more common arcs. If there is at least one common arc between the (i, j) pair of DO, then these DO are grouped together in a temporary buffer set B_i. Finally, all sets B_i are checked to examine the existence of common

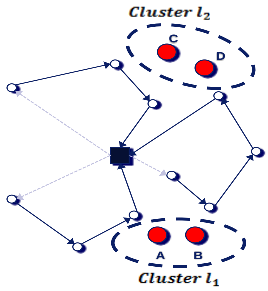
DO; if two buffer sets B_i and B_j contain at least one common DO, then they are joined to a cluster. This procedure terminates with the creation of Λ clusters.

Step 2.3. If there is $n \in \Lambda$ for which $|l_n| > 6,$, then the last arc stored in each arc-set S_i, of all DO $i \in F$ is discarded (i.e. $|S_i| \leftarrow |S_i| - 1, \forall i \in F$), and step 2.2 is repeated. Once $|l_n| \leq 6, \forall n \in \Lambda$, continue to Step 3.

<u>Step 3: Final Solution</u>

The sequence of selecting clusters to be examined for DO insertion affects the overall solution. For that reason, all possible combinations of selecting clusters are checked (*primary combinations*). When intractable, i.e. $\Lambda > 6$, 100 random sequences of clusters are used. For each cluster, all possible insertion combinations of the available DO in this cluster are checked (*secondary combinations*), and the best one is implemented. Figure 6.20 illustrates the above process.

Example



Primary Combinations	Secondary Combinations
$\{l_1\ l_2\} \rightarrow \{[A, B]\ [C, D]\}$	$A \rightarrow B \rightarrow C \rightarrow D$ $B \rightarrow A \rightarrow C \rightarrow D$ $A \rightarrow B \rightarrow D \rightarrow C$ $B \rightarrow A \rightarrow D \rightarrow C$
$\{l_2\ l_1\} \rightarrow \{[C, D]\ [A, B]\}$	$C \rightarrow D \rightarrow A \rightarrow B$ $D \rightarrow C \rightarrow A \rightarrow B$ $C \rightarrow D \rightarrow B \rightarrow A$ $D \rightarrow C \rightarrow B \rightarrow A$

Figure 6.20. Example of considering combinations in the local update heuristic

<u>Specifically</u>:

Step 3.1. Enumerate all primary combinations of clusters, denoted as $f = 1,2, \dots, \mathcal{L}$. Begin with $f = 1$ and execute Steps 3.2. to 3.6.

Step 3.2. For each cluster l_n of DO in primary combination f, determine all possible insertion combinations of competing DO, denoted as C^{l_n} (called hereafter secondary combinations). Initialize the procedure with $n = 1$ and proceed to Step 3.3.

Step 3.3. For each secondary combination $\mathsf{C}_k^{l_n}$, where k corresponds to the secondary combination examined (i.e. $k = 1,2, \dots, |\mathsf{C}^{l_n}|$), apply an insertion heuristic to insert all DO in the order they appear in $\mathsf{C}_k^{l_n}$. For successful insertion of a DO, all problem constraints should be satisfied. In case of any violation, the corresponding DO is not inserted. After each insertion, apply a $2-\text{opt}$ post-optimization procedure. After completing the insertion of all feasible orders in $\mathsf{C}_k^{l_n}$, compute the cost and number of serviced DO for this secondary combination,

$c(C_k^{l_n})$ and $w(C_k^{l_n})$, respectively. Repeat this procedure for all combinations $C_k^{l_n}$ for cluster l_n and proceed to step 3.4.

Step 3.4. Amongst the secondary combinations of cluster l_n, select the one that serves the maximum number of DO. If there are two or more combinations with the same number of DO, then select the one with the minimum cost. We refer to this combination as $\tilde{C}^{l_n}$.

Step 3.5. If all clusters are checked, i.e. $n = \Lambda$, go to Step 3.6, otherwise $n = n + 1$ and go to Step 3.2.

Step 3.6. Compute the final number of served DO and the associated total cost of primary combination f, i.e. $W_f = \sum_{i=1}^{L} w(\tilde{C}^{l_i})$ and $C_f = \sum_{i=1}^{L} c(\tilde{C}^{l_i})$

Step 3.7. If $f < \mathcal{L}$, set $f = f + 1$ and go back to Step 3.2. Otherwise, terminate the procedure and determine the solution, or solutions, $\tilde{f}_w$ that serve the maximum number of DO. From those solutions, implement solution $\tilde{f}$ with the lowest routing cost C_{final} over all solutions $\tilde{f}_w$.

In case the final solution contains DO that may not be inserted, these orphan requests will be added to a pool to be checked for insertion, along with the newly arrived requests, in the next re-optimization cycle.

6.5.4 Computational results

In this Section, we apply the proposed B&P method on the MADREL data for DO planning. We compare the results obtained with: a) the current practice, in which planners assign manually the DO and b) the insertion heuristic presented in Section 6.5.3. The re-optimization period is 1 hour, and we employ the PR tactic.

In addition to DO planning, the case study also analyzed the performance of the algorithms when a commercial software is used to plan the initial routes for the static orders (SO). Thus, the experimental analysis involved the variants presented in Table 6.6.

Table 6.6. Components involved in the experimental investigation

Planning Variants	Involved in	Description
Manual	Planning SO, DO	The current manual process followed by the dispatchers of the courier company; it involves planning SO and DO
SW	Planning SO	Initial routing of SO using a commercial routing software
HEUR	Planning DO	The MADREL insertion heuristic
B&P	Planning DO	The proposed branch-and-price method

Table 6.7 provides information regarding the testing scenarios. *S0* corresponds to the baseline comprising manual planning for DO and SO; *S1* and *S2* employ manual planning for SO and the MADREL heuristic (HEUR) or the proposed B&P algorithm, respectively, for DO planning; in *S3* and *S4* the commercial software is used for SO planning and, the heuristic or the B&P for DO planning.

Table 6.7. Planning scenarios

Scenario	**SO planning**		**DO planning**		
	Manual	**SW**	**Manual**	**HEUR**	**B&P**
S0	✓		✓		
S1	✓			✓	
S2	✓				✓
S3		✓		✓	
S4		✓			✓

Table 6.8 presents the overall results obtained for the 3-day period. The first column notes the five scenarios as indicated in Table 6.7. In the following columns, key information regarding the results obtained under each scenario is presented for each day. In particular, for each day, we report: i) $\breve{D}_s$, which denotes the total duration (in hours) of the initial routing plan (assignment of SO to routes), ii) $\breve{D}_d$, which denotes the excess trip duration (in hours) that is due to the insertion of the DO, and iii) the TD, which corresponds to the total (final) duration of all trips.

Table 6.8. Total routing results (in hours) for all scenarios on the 3-day period

Scenario	**Day 1**			**Day 2**			**Day 3**		
	$\breve{D}_s$	$\breve{D}_d$	TD	$\breve{D}_s$	$\breve{D}_d$	TD	$\breve{D}_s$	$\breve{D}_d$	TD
S0	87,8	14,6	*102,4*	90,2	15,1	*105,2*	70,3	14,2	*84,5*
S1	87,8	9,2	*97,0*	90,2	10,6	*100,8*	70,3	9,6	*79,9*
S2	87,8	6,9	*94,7*	90,2	5,3	*95,5*	70,3	7,3	*77,6*
S3	73,1	14,3	*87,4*	75,4	12,6	*88,1*	60,0	12,3	*72,2*
S4	73,1	11,9	*85,0*	75,4	11,7	*87,2*	60,0	10,8	*70,8*

Figure 6.21 presents the average performance of all scenarios during the 3-day period with respect to $\breve{D}_s$ and $\breve{D}_d$. It is clear that S4 outperforms all other scenarios, reaching an improvement of around 16% on the total trip duration compared to the current planning practices of the courier operator (baseline scenario – S0). The proposed B&P heuristic (S2) outperforms the insertion heuristic (S1) on all days of the investigated period. The B&P algorithm outperforms the insertion heuristic by 33.8% on the average in terms of the additional

cost (excess cost), i.e. the cost above the initial routing cost. This saving is decreased to 12.2% when SO planning is undertaken by the commercial software. Additionally, employing a commercial software for the planning phase may yield an average of 16% improvement compared to the manual processes. An interesting observation is that optimal (or near-optimal) SO plans lead to considerably higher excess costs for the DO (as shown in S3 and S4). This may be due to the fact that including DO in the initial optimal routes results to significant deviations from these routes.

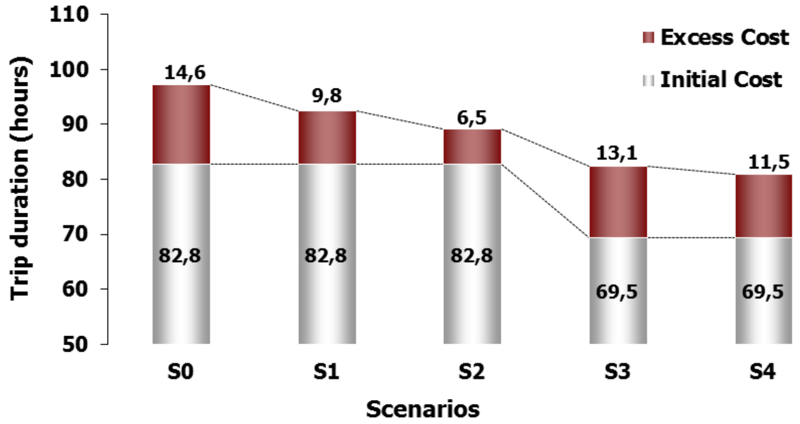


Figure 6.21. Overall performance of all scenarios (average of all days)

Figure 6.22 presents the average working time per vehicle per day. It also shows the average improvement in vehicle productivity; that is $\%100|(T_{S0} - T_{Sx})/T_{S0}|$, where T_{S0} is to the overall trip duration per vehicle of S0 and T_{Sx} the trip duration of scenarios Sx, where $x \in \{1,2,3,4\}$. The Figure shows that employing sophisticated tools for SO and DO planning can offer to the courier service provider savings of up to 17% per driver shift (i.e. around 1.3 hours on an 8-hour working shift).

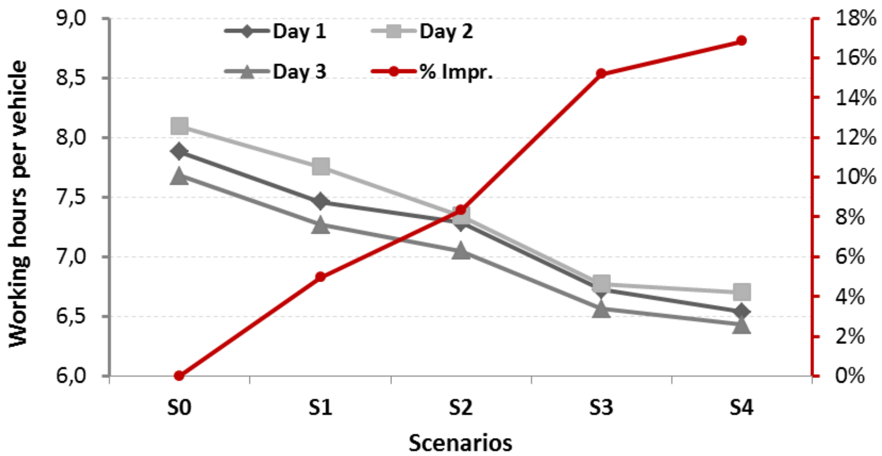


Figure 6.22. Overall performance of scenarios w.r.t. the working hours per vehicle

Finally, Figure 6.23 illustrates the improvement of the cost per routed request (for both SO and DO), as well as the cost per DO for all scenarios. The y-axis provides the percentage difference (improvement) of the cost under each scenario over S0. The cost is calculated by dividing the TD (see Table 6.7) with the total number of routed requests. As stated earlier, S4 results to the best overall unit cost performance (the unit cost per request improves by almost 18%). The improvement of the unit cost per DO for scenarios S1 and S2 is up to 56%, which illustrates the efficiency of applying sophisticated methods for the allocation of DO.

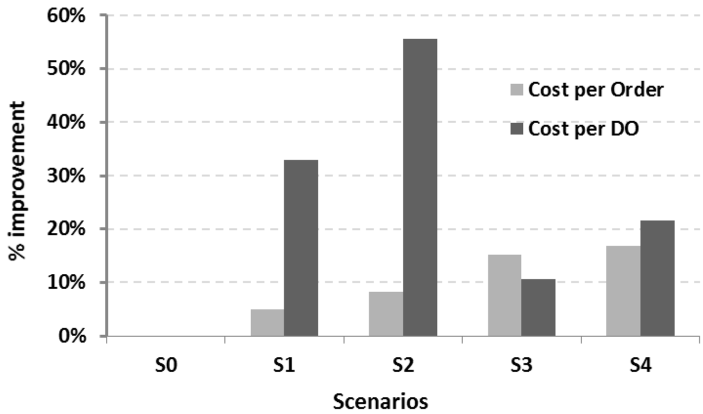


Figure 6.23. Average performance of scenarios w.r.t. unit cost (cost per request)

Chapter 7: THE DVRPMB WITH LOAD TRANSFERS

7.1 Introduction

We have already studied DVRPMB by considering that static delivery orders originally assigned to vehicles cannot be re-allocated to other vehicles, while dynamic orders may be served by any vehicle as needed. However, maintaining the original assignment of delivery orders to vehicles may limit system performance, since the changes in the system state caused by the arriving dynamic orders may grant re-assignments of such orders advantageous. Thus, in this Chapter, we introduce and solve a variant of DVRPMB that allows orders to be transferred between vehicles during plan implementation. This significant differentiation from DVRMB introduces considerable complexity that needs to be dealt in a fundamentally different way. We refer to this problem as the ***DVRPMB with Load Transfers (DVRPMB-LT)***. By allowing load transfers between vehicles, we attempt to better utilize the fleet by re-distributing its workload as needed based on the dynamic state of the system.

A significant observation from the analysis of typical DVRPMB settings that prompted this work is that the original assignments of static delivery orders to vehicles may result in significant *overlaps* of vehicle routes in the solution of the re-optimization problem. These overlaps are caused by the dynamic pick-up orders (DO), and, by definition, increase costs (see Figure 7.1). In general, we have identified two (2) main conditions under which such undesirable overlaps might occur:

a) New vehicles may be dispatched from the depot to serve newly received dynamic orders; in those cases, route overlaps are possible
b) A vehicle that has completed its tasks at an early stage may be assigned to serve newly received dynamic orders resulting to overlaps.

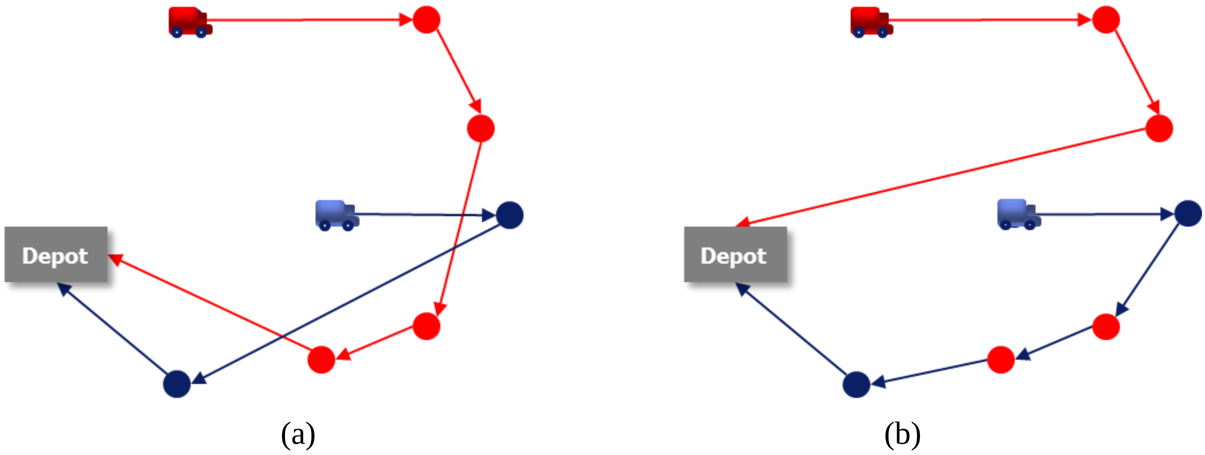


Figure 7.1. Example of (a) overlapping, and (b) non-overlapping vehicle routes

It is noted that urban logistics companies use load transfer practices to facilitate their distribution operations. Specifically, certain courier companies employ real-time load-transfers especially in cases in which the service area has been partitioned into a number of geographic zones (regions) and each vehicle (driver) is tasked to work within the boundaries of such a zone. If an order (e.g. package) is picked up from a location within a certain zone and needs to be delivered to a different zone, the drivers communicate and decide where and when to meet in order to transfer the corresponding order. In some cases, there are predefined locations where this operation may be performed, usually referred to as “transshipment points” (Mitrovic-Minic and Laporte, 2006).

In addition, this practice is also met in money-transfer operations. In those settings, armored vehicles executing a distribution plan are called to serve ATM requests for money collection (or service) that arrive to a dispatch center in a dynamic fashion. A unique (physical) key exists per ATM that allows drivers to access it; no other driver is allowed to access an ATM unless she/he holds its key. Typically, drivers are given the keys for all ATMs of their responsible area at the beginning of the day. However, the arrival of dynamic requests disrupts the predefined plan and may result to delays on the agreed time windows or inability of the vehicle covering a certain region to serve all dynamic orders. In those cases, drivers can meet during execution and exchange keys, in order to better re-distribute the work and lower costs.

To the best of our knowledge, this is the first study that addresses transfer operations in such a context. As in DVRPMB, we deal with the DVRPMB-LT by updating (re-optimizing) the *a priori* plan at certain points in time during execution, in order to incorporate the dynamic orders received up to that point. We model the underlying re-optimization problem using an arc-based and we compare the exact solutions obtained to the exact solutions of DVRPMB, which does not allow transfers. Furthermore, we develop a practical heuristic framework in order to address the complexity of DVRPMB-LT and solve cases of practical size. Subsequently, we employ the proposed framework to solve and analyze the full dynamic problem, and investigate the impact of different re-optimization policies on the solution quality.

The remainder of this Chapter is structured as follows: Section 7.2 overviews the most relevant approaches in the literature regarding transfer (or transshipment) operations in VRPs. Section 7.3 describes the problem setting and formalizes the re-optimization problem with load transfers. Section 7.4 introduces the proposed heuristic solution framework to solve the re-optimization problem for instances of practical size. Finally, Section 7.5 presents computational results for both the re-optimization problem and the overall dynamic problem. The solutions obtained are also compared to the solutions of DVRPMB without load transfers.

7.2 Related literature on transfer (transshipment) operations

The re-optimization problem of DVRPMB-LT is relevant to the PDP with Transfers (PDPT, Cortes *et al.*; 2010), the PDP with Time Windows and Transshipment (PDPTWT, Mitrovic-Minic and Laporte; 2006), and the DARP with Transfers (DARPT, Masson; 2014). In those problems, goods or passengers are associated with a pick-up and a delivery location and may be transshipped at pre-specified locations; i.e., vehicles are allowed to drop goods or persons temporarily so that they are picked up and delivered to the final destination by another vehicle. Other related environments where transfer operations have been studied include the school bus routing problem (Nakao and Nagamochi, 2008), the robotized pick-up and delivery process of items requested by users in an office building (Coltin and Veloso, 2012), and environments related to supply chain decisions (Dondo *et al.*, 2009), or cross-docking operations (Petersen and Ropke, 2011).

In the following paragraphs we review work related to the PDPT and PDPTWT. Table 7.1 summarizes information on interesting references related to transfer operations.

Shangh and Cuff (1996) were the first to discuss the PDPT. The authors employ a look-ahead heuristic approach for picking up and delivering patient records, equipment and supplies for a health maintenance organization (HMO). The heuristic constructs mini-routes and assigns them to vehicles. Transfers are only considered when an order cannot be inserted in the current solution without adding an extra vehicle. The dynamic version of a similar problem is considered by Thangiah *et al.* (2007) who improved the results of Shangh and Cuff (1996) by incorporating a local search phase. Mitrovic-Minic and Laporte (2006) studied the PDPTWT motivated by a large San Francisco-based courier company that uses transshipment of loads between vehicles. A single transshipment is allowed per request and up to 4 locations were considered as potential fixed transfer locations. The authors proposed a two-phase heuristic to solve generated instances with up to 100 orders, and demonstrated that transshipment operations can significantly reduce the total distance traveled by vehicles, especially in clustered cases.

Mues and Pickl (2005) proposed a column generation-based heuristic for the PDPT with a single fixed transfer location. They evaluated their algorithm considering instances of up to 70 orders. Gørtz *et al.* (2008) considered a version of PDPT, and proposed heuristics for the capacitated and uncapacitated cases in order to minimize the maximum completion time (makespan) of operations. Petersen and Ropke (2011) considered a case of pick-up and delivery of flowers in Denmark with a single fixed transfer location. They proposed an Adaptive Large Neighborhood Search (ALNS) algorithm, which they applied to practical instances of up to 982 orders. For the PDPT, Qu and Bard (2012) also proposed an ALNS within a greedy randomized adaptive search procedure (GRASP) framework. They applied their method to instances with up to 25 orders, obtaining solutions within 1% of the optimal ones. Masson *et al.* (2011) also proposed an ALNS algorithm for the PDPT and reported competitive results for the Mitrovic-Minic and Laporte (2006) instances, and for practical instances with up to 193 orders. Masson *et al.* (2014) extended the ALNS technique in order to solve the Dial-a-Ride Problem with Transfers (DARPT). They reported savings up to 8% due to the introduction of transfer operations. Lin (2008) presented a PDPTWT in which all requests share the same delivery location (but delivery time windows are different), and a transfer can occur at the last pick-up before a delivery. The authors presented an integer programming formulation and were able to solve instances of up to 100 orders using a commercial solver.

Few exact approaches exist for the PDPT. Cortes *et al.* (2010) introduced an arc-based formulation by considering fixed transfer locations. They employed a Branch-and-Cut algorithm using Benders Decomposition and were able to solve to optimality instances with up to six orders and two vehicles, reporting superior computational performance compared to a standard Branch-and-Bound technique. Kerivin *et al.* (2008) presented a Branch-and-Cut algorithm in order to solve a PDPT without time windows, in which every order can be transferred from one vehicle to another at every node of the network. The authors were able to solve instances with up to 15 orders. Nakao and Nagamochi (2008) presented a lower bound calculation for the PDPT with a single transfer location and no time windows.

Table 7.1. Key information in references investigating transshipment operations

Reference	Problem	Capacity	TW	Environ-ment	Transfer Locations	Solution Procedure
Shangh and Cuff (1996)	1-1 P&D of medical equipment	-	✓	Static	All problem nodes	Look-ahead insertion heuristic
Mues and Pickl (2005)	1-1 P&D of freight	-	✓	Static	Single location	Column-generation
Mitrovic-Minic and Laporte (2006)	1-1 P&D of parcels and letters	-	✓	Static	Predefined locations	Two-phase local search
Thangiah *et al.* (2007)	1-1 P&D of freight or passengers	-	✓	Dynamic	All problem nodes	Heuristic
Lin (2008)	1-1 P&D of parcels and letters	-	✓	Static	Special customer locations	Integer programming
Kerivin *et al.* (2008)	1-1 P&D of freight or passengers	-	-	Static	All problem nodes	Mixed-integer linear programming
Nakao and Nagamochi (2008)	School bus routing problem	-	-	Static	Single location	-
Gørtz *et al.* (2009)	Dial-a-ride	✓	-	Static	All problem nodes	Heuristic
Cortes *et al.* (2010)	Dial-a-ride	✓	✓	Static	Predefined locations	Branch-and-Cut
Masson *et al.* (2011)	1-1 P&D of freight or passengers	✓	✓	Static	Predefined locations	ALNS
Petersen and Ropke (2011)	1-1 P&D of flower containers	✓	✓	Static	Single location	ALNS
Masson *et al.* (2014)	Dial-a-ride	✓	✓	Static	Predefined locations	ALNS
Qu and Bard (2012)	1-1 P&D of freight	✓	✓	Static	Single location	GRASP combined with ALNS

We differentiate our current work in the following three aspects: first, to the best of our knowledge, this is the first study that introduces transfer operations in the 1-M-1 PDPs, in which orders are not associated to a pick-up and delivery pair, but to a single location, either pick-up or delivery. Second, in this work we introduce transfer operations in a dynamic environment in which we investigate how transfer practices affect the solution of the overall dynamic problem with respect to different frequencies of re-optimization. Finally, the majority of the related work considers fixed (predefined) locations, in which transfers are allowed, whereas in this study we investigate additional options for allowing transfer operations to take place at all nodes of the network. The latter has been investigated by limited number of studies (see Table 7.1).

7.3 Re-optimization in DVRPMB-LT

7.3.1 The role of re-optimization in solving the DVRPMB-LT

As already mentioned, DVRPMB-LT employs the DVRPMB setting described in previous chapters and is dealt through iterative re-optimization (see also Figure 7.2). However, in DVRPMB-LT a vehicle $k \in K$ is allowed to serve delivery (static) orders assigned to another vehicle $k' \in K, k' \neq k$ during the solution of the re-optimization problem, provided that the required order transfer is feasible. This makes the re-optimization problem quite different from that of DVRMB and quite interesting. In addition, it introduces significant complexity that needs to be dealt in a fundamentally different way.

Figure 7.2. The re-optimization process

The setting and the formulation of the *re-optimization problem with transfers*, henceforth denoted as **DVRPMB-LT**($\boldsymbol{\ell}$), is described below (Sections 7.3.2 and 7.3.3). Consequently, its solution approach is described in Section 7.4.

It should be noted, however, that the solution strategy for the DVRPMB-LT via re-optimization requires also the definition of the appropriate re-optimization policy and tactic. Regarding the former, we investigate in Section 7.5.4 how different policies affect the solution of DVRPB-LT. Regarding the latter (implementation tactic), we investigate DVRPMB-LT under the PR policy, since our extensive experiments in Chapters 5 and 6 indicated that it is the most promising tactic.

7.3.2 Basic assumptions of the re-optimization problem DVRPMB-LT($\boldsymbol{\ell}$)

Allowing load transfers may raise significantly the operational complexity of a logistics system. For example, it may not be practical from a management perspective to allow multiple transfers per order, or a vehicle to exchange orders with more than one vehicle(s). Such practices may confuse both drivers and dispatchers, and lead to excessive managerial overheads. Taking into consideration this operational issue, below we define a set of assumptions within which load transfers are practical and possible. Note that transfer operations are relevant to delivery (static) orders only, since pick-up orders (DO) can be collected by any vehicle. The operating assumptions are as follows:

d) All orders need to be satisfied (both delivery and pick-up ones)
e) For the re-optimization problem, each vehicle is allowed to participate in only one transfer operation throughout its remaining (not executed) route prescribed by the revised plan. Of course, a vehicle may participate in more than one transfer operations during its entire executed route (multiple re-optimizations).
f) With respect to transfer locations, we investigate two cases; vehicles may meet and transfer loads: i) at fixed (predefined) locations known prior to the start of operations, or ii) at all not yet served customer locations (including current vehicle locations and those of dynamic order clients).

Regarding the third assumption, it should be noted that fixed transfer locations are typically predefined facilities dispersed throughout the distribution area, where a vehicle is able to discharge load that may be later picked-up by a different vehicle. In this fixed location, vehicles are not required to be physically present at the same time. On the other hand, when transfers are allowed at the location of any not yet served customer, vehicles have to be physically present at the same location (even if that means that one of the two vehicles will have to wait for the other one to arrive).

From a business perspective, the second assumption (*one-to-one* transfer policy) is practical, streamlining fleet management and driver overhead. From an algorithmic perspective, this assumption limits the problem's complexity significantly and allows the re-optimization problem to be considered as the combination of pairwise sub-problems, as will be described in Section 7.4.

In general, a feasible solution of DVRPMB-LT(ℓ) should satisfy the following:

i) All vehicle routes should start at the current vehicle location and finish at the depot (no cycles)
ii) All customer nodes (delivery orders and DO) must be served and should be visited exactly once (note that in Section 7.3.3 we introduce additional nodes where transfer takes place, thus customer nodes will be always visited once).
iii) If an order is to be transferred from vehicle $k \in K$ to vehicle $k' \in K$, then vehicle k should arrive at the transfer location prior to the departure of vehicle k'
iv) Each vehicle should participate in a single transfer operation at most

v) The load of each customer has to be transported by a single vehicle at any time

vi) All customers should be served within their associated time windows (if any)

vii) Vehicles should return to the depot prior to the end of the working horizon (T_{max})

viii) The vehicle capacity cannot be exceeded.

The problem's objective is to minimize the routing costs (i.e. distance traveled), subject to constraints (i)-(viii) above. We should note here that load transfers might cause additional delays due to transfer operations (e.g., onsite time to load/unload items, waiting for the other vehicle to arrive, etc.). Within our setting, we do not attempt to minimize such delays in the objective function; however, such delays are considered by the problem constraints.

An illustrative example

Consider the example of Figure 7.3 for a case with two vehicles. Initially, the two vehicles are set to serve delivery (static) orders only. At the re-optimization timestamp, the vehicles have already executed a portion of their planned routes and are currently located at customers 2 and 6, respectively (Figure 7.3a). Furthermore, a number of DO have been received and are to be incorporated in the current plan. Figure 7.3b illustrates the results of the DVRPMB re-optimization algorithm of Chapter 5 with all DO been assigned to vehicles. The total routing cost is 108.1. Given this solution, we examine the possible efficiencies from load transfers.

Figure 7.3c illustrates a possible solution, assuming that transfer operations are allowed only at a predefined (fixed) location with coordinates (15,0). This solution includes the transfer of orders 7 and 8 from the red to the blue vehicle; the total routing cost is 86.4, resulting in approximately 20% savings. Figure 7.3d illustrates a solution in which transfer is allowed at the location of any not yet served customer. The two vehicles meet at the location of customer 10, where the red vehicle will again transfer customer orders 7 and 8 to the blue vehicle. This solution results to a cost of 84.5 and total savings of 21.9% compared to the solution of Figure 7.3b.

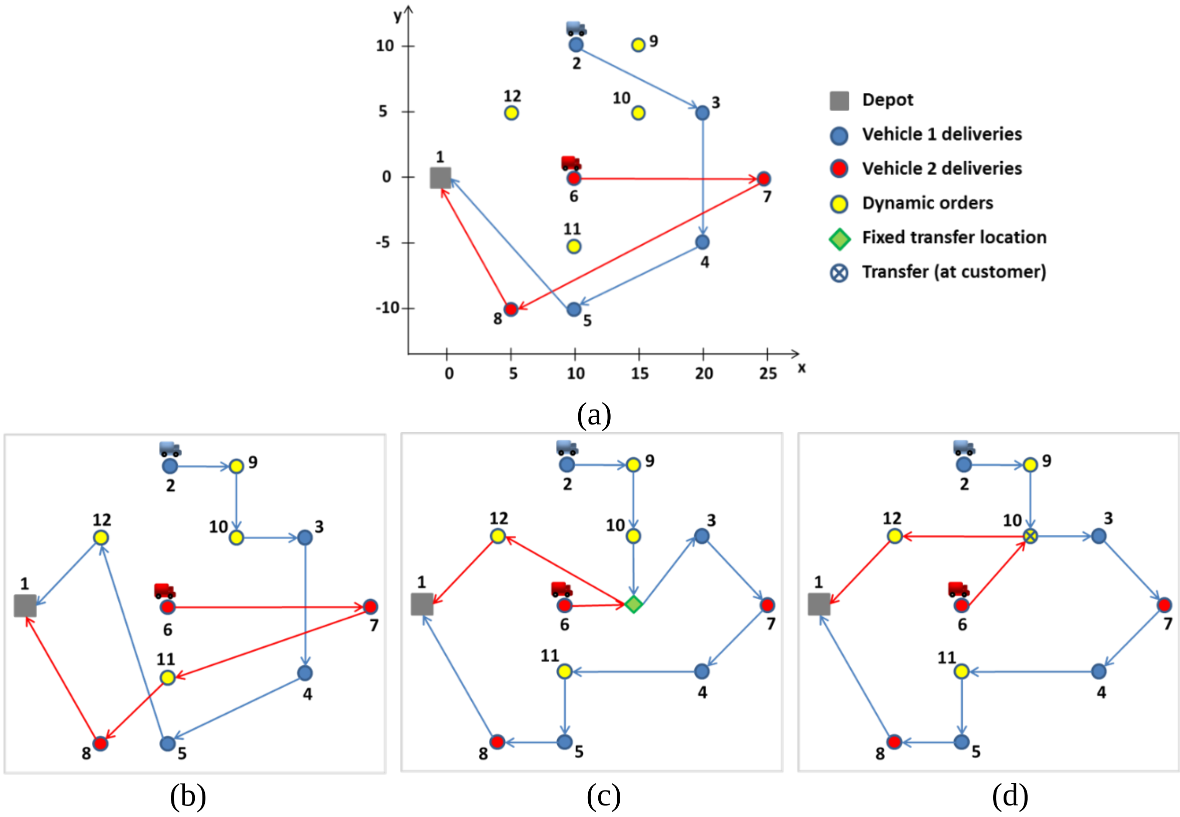


Figure 7.3. Illustrative example for the DVRPMB-LT(ℓ); (a) routes prior to re-optimization, (b) solution of DVRPMB without transfers, (c) load transfers at fixed location; (d) load transfers at customer location

7.3.3 Mathematical formulation of DVRPMB-LT(ℓ)

The proposed formulation has been based on the work of Cortes *et al.* (2010), in which the authors present a Mixed Integer Linear Programming (MILP) model for the Pickup and Delivery Problem with Transfers (PDPT). In the current work, we extend and adjust this formulation to be able to consider: i) pick-up and delivery orders which are not paired, and ii) potential locations for transferring loads to be all not yet served customer locations (including current locations of vehicles).

7.3.3.1 Modelling assumptions

In the classical DVRPMB setting, each customer node is associated only with a single type of operation, which is either unloading (delivery orders) or loading (pick-up orders), but not both. The setting of DVRPMB-LT uses the concept of transfer locations, where vehicles may load and unload goods. These locations may be client locations, or special pre-designated locations in the operational area. As in the work of Cortes *et al.* (2010), in order to capture the difference between operations (load/unload), every transfer location u is split into two separate nodes, $s(u)$ and $f(u)$, which correspond to the start and finish of the transfer operation, respectively (see Figure 7.4). When a vehicle enters a transfer location u, it initially enters node $s(u)$ to unload the orders to be transferred to another vehicle (if any). The vehicle then proceeds (notionally) to node $f(u)$, where orders (dropped at node $s(u)$by another vehicle) may be waiting to be loaded.

Figure 7.4. Representation of the transfer location

To consider transfers at customer locations, we define two additional sets of nodes M' and N' to duplicate the sets containing the current vehicle locations (set M) and the customer nodes (set N), respectively. We also define as $0'$ the transfer location corresponding to the depot. Those duplicate nodes will participate in the set of possible transfer locations (set U, see Table 7.2). Thus, each location of not yet served customer is represented by three (3) distinct nodes, namely: a) the original node $i \in (N \cup M \cup 0)$, b) the start transfer node $s(u)$, and c) the finish transfer node $f(u)$, where $u \in (N' \cup M' \cup 0')$ denotes the transfer node associated with node $i \in (N \cup M \cup 0)$. Note that all three nodes are considered to be at the same geographical location (and the distances between them are equal to zero).

In case the transfer location corresponds to a customer location $i \in N$, the vehicle first visits (and serves) the customer node, then proceeds to node $s(u)$ to begin the transfer operation (unload) and finally proceeds to node $f(u)$ (in order to exit the transfer location). The second vehicle, which participates in the transfer operation but does not serve node i arrives directly to node $s(u)$ (and immediately moves to node $f(u)$ to reload). This operation is modeled as shown in Figure 7.5.

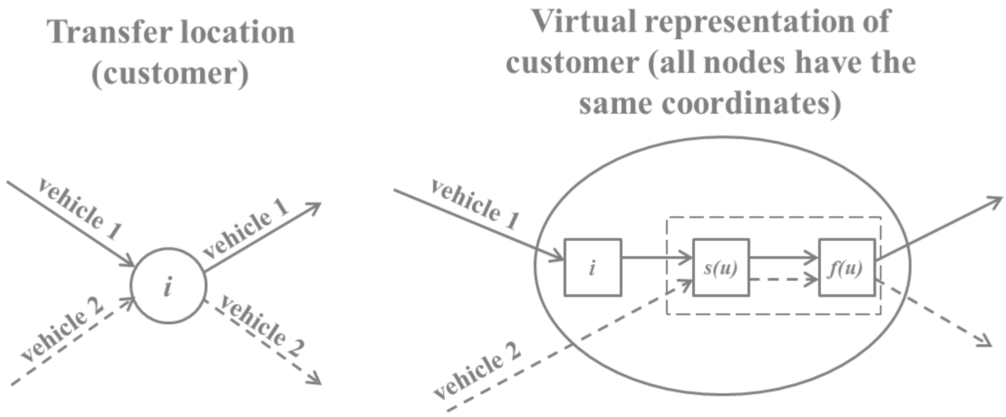


Figure 7.5. Modeling the case of transferring loads at customer location $\boldsymbol{i}$

7.3.3.2 Mathematical model

Below we present the mathematical model for the re-optimization problem (DVRPMB-LT(ℓ)). It should be noted that this formulation is able to solve instances with K vehicles with the assumption that each vehicle can participate in only one transfer operation throughout its remaining route.

Prior to presenting the mathematical model for the DVRPMB-LT(ℓ), we first summarize the notation involved in the formulation (see Table 7.2). Additionally, we define the set of links involved in the formulation, i.e. we exclude edges (arcs) that are not reasonable within the context of DVRPMB-LT, as for example direct links from the start transfer nodes $s(u), u \in U$ to any customer node $i \in N$. Let A be the set of arcs, with $A = A_1 \cup A_2 \cup A_3 \cup A_4$, where (see also Figure 7.6):

$A_1 = \{\{\mu_k : k \in K\} \times (N \cup s(U) \cup \{0\})\}$
$A_2 = \left(N \times (N \cup s(U) \cup \{0\})\right) \setminus \{(i,i) : i \in N\}$
$A_3 = \left\{\left(s(u), f(u)\right) : u \in U\right\}$
$A_4 = (f(U) \times (N \cup \{0\} \cup s(U))) \setminus \{(f(u), s(u)) : u \in U\}$

Table 7.2. Notation for DVRPMB-LT($\boldsymbol{\ell}$)

Notation	Description	Comment
K	Set of vehicles *en route*[8]	
M	Starting location of vehicles K	$M = \bigcup_{k \in K} \{\mu_k\}$
C	Set of nodes associated to committed (delivery) orders	$C = \bigcup_{k \in K} C_k$
F	Set of nodes of flexible (pick-up) orders – DO	
N	Set of nodes associated with customer orders	$N = C \cup F$
0	Depot location	
U_f	Set of fixed transfer location node(s)	
M'	Set of transfer locations corresponding to starting location of vehicles	
N'	Set of transfer locations corresponding to customer nodes	
$0'$	Transfer location corresponding to the depot (may be used for intermediate exchange)	
U	Set of all transfer location nodes	$U = U_f \cup \{0'\} \cup M' \cup N'$
$s(u)$	Start node of transfer location $u \in U$	
$f(u)$	Finish node of transfer location $u \in U$	
$s(U)$	Set of all start nodes of transfer locations	$s(U) = \{s(u) : u \in U\}$
$f(U)$	Set of all finish nodes of transfer locations	$f(U) = \{f(u) : u \in U\}$
W	Set of all nodes	$W = N \cup M \cup \{0\} \cup s(U) \cup f(U)$

[8] Note that no vehicles are assumed available at depot

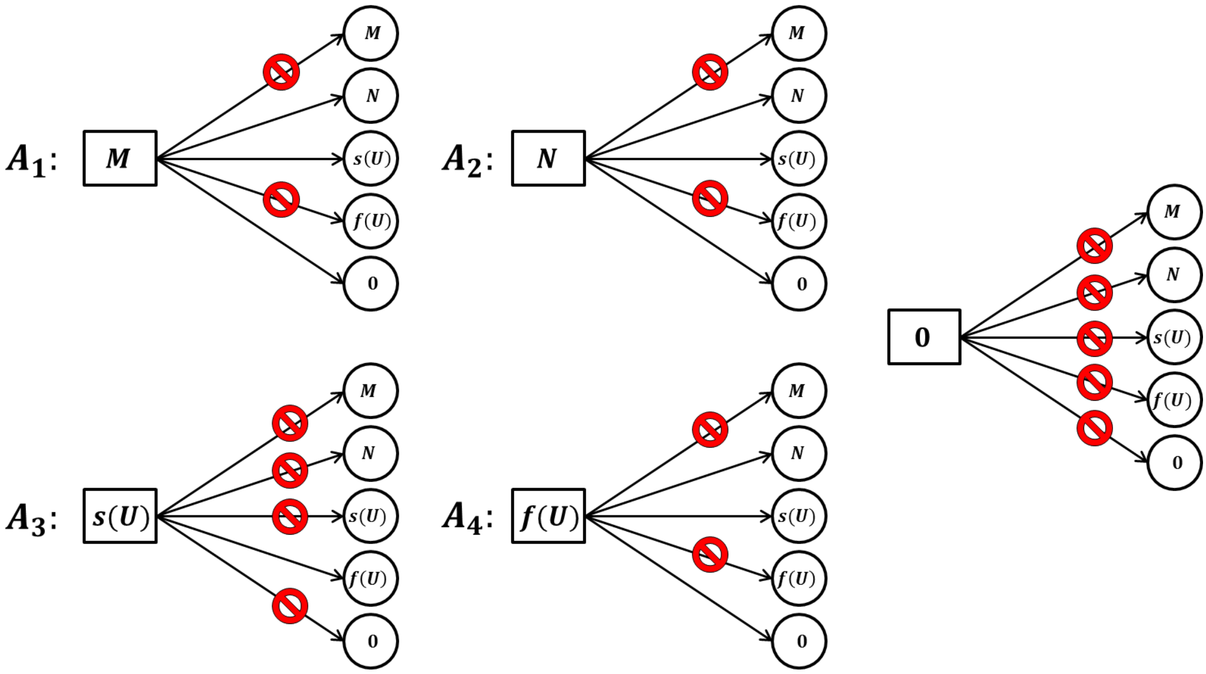


Figure 7.6. Allowable arcs $(\boldsymbol{i}, \boldsymbol{j}) \in \boldsymbol{A}$ (possible connections from any $\boldsymbol{i}$ to any $\boldsymbol{j}$, $\forall \boldsymbol{i}, \boldsymbol{j} \in \boldsymbol{W}$)

We also denote c_{ij} and t_{ij} to be the travel cost and travel time corresponding to arc $(i, j) \in A$, respectively. Finally, recall that each node $i \in C \cup F$ is related to a demand/supply value d_i and requires service within time window $[a_i, b_i]$, with a service duration s_i.

The proposed mathematical formulation involves three (3) types of decision variables: a) binary variables x_{ijk} which are used to model the vehicle routes and are equal to 1 if arc $(i, j) \in A$ is transversed by vehicle $k \in K$ and zero otherwise; b) binary variables z_j^{ki} which are used to and keep track of the status of each customer order while it is traveling from node to node, as in the formulation of Cortes *et al.* (2010). These variables are equal to 1 if customer order $i \in N$ is onboard vehicle $k \in K$ when it arrives to node $j \in W \backslash M$, and 0 otherwise, for all $i \in N, k \in K$; c) finally, real variables w_{ik} are associated with the arrival time of vehicle $k \in K$ at each node $i \in W$; accordingly, $w_{s(u)k}$ and $w_{f(u)k}$ correspond to the time of arrival and time of departure of vehicle $k \in K$ to/from the transfer location, respectively.

The objective of the re-optimization problem is to minimize the total cumulative routing cost over the planning horizon $[\mathrm{T}_\ell, \mathrm{T}_{\max}]$ and is given by:

$$min(z) = \sum_{k \in V} \sum_{(i,j) \in A} c_{ijk} x_{ijk} \tag{7.1}$$

Route constraints

$$\sum_{j\in W\setminus(M\cup f(U))} x_{\mu_k jk} = 1 \qquad \forall k \in K \tag{7.2}$$

$$\sum_{i\in N\cup\{\mu_k\}\cup f(U)} x_{i0k} = 1 \qquad \forall k \in K \tag{7.3}$$

$$\sum_{i\in W\setminus(\{0\}\cup s(U))} x_{ihk} - \sum_{j\in W\setminus(M\cup f(U))} x_{hjk} = 0 \qquad \forall k \in K, \forall h \in N \tag{7.4}$$

$$\sum_{i\in N\cup\{\mu_k\}} x_{is(u)k} - x_{s(u)f(u)k} = 0 \qquad \forall k \in K, \forall u \in U \tag{7.5}$$

$$\sum_{j\in N\cup\{0\}} x_{f(u)jk} - x_{s(u)f(u)k} = 0 \qquad \forall k \in K, \forall u \in U \tag{7.6}$$

Customer constraints

$$\sum_{k\in K}\sum_{j\in W\setminus(M\cup f(U))} x_{ijk} = 1 \qquad \forall i \in N \tag{7.7}$$

Time-based constraints

$$x_{\mu_k ik} = 1 \implies w_{ik} \geq t_{\mu_k i} \qquad \forall k \in K, \forall i \in N \cup 0 \tag{7.8}$$

$$x_{\mu_k s(u)k} = 1 \implies w_{s(u)k} \geq t_{\mu_k s(u)} \qquad \forall k \in K, \forall u \in U \tag{7.9}$$

$$x_{ijk} = 1 \implies w_{jk} \geq w_{ik} + t_{ij} + s_i \qquad \forall k \in K, \forall (i,j) \in \{(i,j): i \in N, j \in N \cup 0\} \tag{7.10}$$

$$x_{is(u)k} = 1 \implies w_{s(u)k} \geq w_{ik} + t_{is(u)} + s_i \qquad \forall k \in K, \forall i \in N, \forall u \in U \tag{7.11}$$

$$x_{s(u)f(u)k} = 1 \implies w_{f(u)k} \geq w_{s(u)k} + t_{s(u)f(u)} \qquad \forall k \in K, \forall u \in U \tag{7.12}$$

$$x_{f(u)jk} = 1 \implies w_{jk} \geq w_{f(u)k} + t_{f(u)j} \qquad \forall k \in K, \forall j \in N \cup 0, \forall u \in U \tag{7.13}$$

$$x_{f(u)s(\varphi)k} = 1 \implies w_{s(\varphi)k} \geq w_{f(u)k} + t_{f(u)s(\varphi)} \qquad \forall k \in K, \forall u \in U, \forall \varphi \in U\setminus\{u\} \tag{7.14}$$

Flow of requests constraints

$$\sum_{k\in K}\sum_{i\in F} z_{\mu_k}^{ki} = \sum_{k\in K}\sum_{i\in C_k} z_{\mu_k}^{ki} - |C| = 0 \tag{7.15}$$

$$\sum_{k\in K}\sum_{i\in C} z_0^{ki} = \sum_{k\in K}\sum_{i\in F} z_0^{ki} - |F| = 0 \tag{7.16}$$

$$x_{hjk} = 1 \implies z_h^{ki} = z_j^{ki} \qquad \forall k \in K, \forall i \in N, \forall (h,j) \in A^{U}[9] \text{ such that } h \neq i \tag{7.17}$$

[9] $A^U = A \setminus \{(s(u), f(u) | u \in U)\}$

$$x_{ijk} = 1 \implies z_i^{ki} - z_j^{ki} = 1 \qquad \forall k \in K, \forall i \in C, \forall j \in W \backslash (M \cup f(U)) \qquad (7.18)$$

$$x_{ijk} = 1 \implies z_j^{ki} - z_i^{ki} = 1 \qquad \forall k \in K, \forall i \in F, \forall j \in W \backslash (M \cup f(U)) \qquad (7.19)$$

$$\sum_{k \in K} z_{s(u)}^{ki} - \sum_{k \in K} z_{f(u)}^{ki} = 0 \qquad \forall u \in U, \forall i \in N \qquad (7.20)$$

$$z_{s(u)}^{ki} + z_{f(u)}^{mi} = 2 \Rightarrow w_{f(u)m} \geq w_{s(u)k} + \tilde{\Xi} \qquad \forall u \in U, \forall k, m \in K, k \neq m, \forall i \in N \qquad (7.21)$$

$$z_{s(u)}^{ki} + z_{f(u)}^{mi} = 2 \Rightarrow w_{f(u)k} \geq w_{s(u)m} \qquad \forall u \in U \backslash U_f \cup \{0'\}, \forall i \in N, \forall k, m \in K, k \neq m \qquad (7.22)$$

<u>Operational constraints</u>

$$\sum_{r \in U} \sum_{k \in K} z_{s(u)}^{ki} \leq 1 \qquad \forall i \in N \qquad (7.23)$$

$$\sum_{i \in W \backslash f(U)} \sum_{u \in U} x_{is(u)k} \leq 1 \qquad \forall k \in K \qquad (7.24)$$

$$max(a_i, T) \sum_{j \in W \backslash (M \cup f(U))} x_{ijk} \leq w_{ik} \leq b_i \sum_{j \in W \backslash (M \cup f(U))} x_{ijk} \qquad \forall k \in K, \forall i \in N \qquad (7.25)$$

$$\sum_{i \in N} q_i z_j^{ki} \leq \bar{Q} \qquad \forall j \in N, \forall k \in K \qquad (7.26)$$

Constraints (7.2) – (7.6) correspond to *basic route constraints*; in particular, Constraints (7.2) and (7.3) ensure that the vehicles will depart from their current locations and will eventually return to the depot; Constraint (7.4) ensures flow conservation at the nodes in set N, while Constraints (7.5) and (7.6) ensure flow conservation at the transfer locations. Note also that those constraints permit vehicles to reach a transfer location at most once.

Constraints (7.7) correspond to *customer constraints*, which ensure that all customer orders will be served and the corresponding customer nodes will be visited exactly once. Constraints (7.8) – (7.14) ensure *time feasibility* of a route, and are used to eliminate subtours (cycles). This set of constraints can be written as linear expressions using the big-M technique (Desrosiers *et al.*, 1995; Desrochers *et al.*, 1988; see Section 7.3.3.3). It should be noted that for Constraint (7.12), the travel time between start and end nodes of the transfer location $t_{s(u)f(u)}$, is considered to be a very small positive number in order to avoid zero-cost cycles.

Constraints (7.15) – (7.23) ensure the *flow of orders*. In particular, Constraints (7.15) and (7.16) define the initial and final loading conditions, respectively; i.e., vehicle $k \in K$ starts from its initial location carrying the C orders assigned to it (since the PR tactic is considered, no F orders are considered at the beginning of re-optimization) and ends at the depot with only F orders on board (no C order should be brought back to depot). Constraint (7.17) ensures load continuity; i.e. the load is only unloaded at the designated customer location (the load of node $i \in N$ will be onboard when the vehicle arrives at customer location $j \in N$ if it is also onboard when the vehicle was at the previous customer location $h \in N$). Constraint (7.18) ensures that a delivery order is unloaded when it reaches the location of the corresponding customer. Similarly, Constraint (7.19) ensures that a pick-up order will be loaded at the appropriate location. Constraint (7.20) refers to the flow conservation of the load variables, i.e. it ensures that a customer order that arrives to a transfer location on any vehicle must leave the transfer location by any vehicle (essentially, with the same vehicle and/or the other vehicle of the pair). Constraint (7.21) ensures that if an order is exchanged between two vehicles at a transfer location (i.e. reaches transfer location with vehicle $k_1 \in K$ and leaves transfer location with vehicle $k_2 \in K, k_2 \neq k_1$), then vehicle k_1 has to arrive to the transfer location prior to the departure of vehicle k_2 from the transfer location; $\tilde{\Xi}$ is a scalar that represents the time needed for the load to remain at the transfer location (till its departure). Furthermore, Constraint (7.22) is similar to Constraint (7.21) but ensures the simultaneous presence of both vehicles at the transfer location for those cases for which the transfer operation takes place at a customer location.

Moreover, we consider additional *operational constraints* in (7.23) to (7.25). Specifically, Constraint (7.23) limits the number of times any customer order may be transferred to at most once, while Constraints (7.24) limit the number of transfers per vehicle (as per our original assumption of Section 3.2). Finally, Constraints (7.25) ensure that the each customer $i \in V$ is served within its time window, and Constraints (7.26) ensure that the load carried on the vehicle must not exceed the vehicle's maximum capacity ($\bar{Q}$). Details regarding the linearization of those constraints are provided below (Section 7.3.1.3).

As mentioned above, here we investigate the case for which all orders may be served by the available fleet. In the case of limited resources, for which it is not necessary to satisfy all flexible (pick-up) orders (as discussed mostly in Chapter 6), Constraint (7.7) can be replaced by Constraints (7.27) and (7.28) below.

$$\sum_{k \in V} \sum_{j \in W \setminus (M \cup f(U))} x_{ijk} = 1 \qquad \forall i \in C \qquad (7.27)$$

$$\sum_{k \in V} \sum_{j \in W \setminus (M \cup f(U))} x_{ijk} \leq 1 \qquad \forall i \in F \qquad (7.28)$$

7.3.3.3 Constraint linearization

Several sets of constraints presented in the above mathematical formulation are expressed by non-linear relationships. Here we present the way these constraints may be linearized in order for the model to be solved by a commercial solver (CPLEX). The linearization was based on the big-M technique (Desrosiers *et al.*, 1995; Desrochers *et al.*, 1988).

Consider the set of Constraints (7.8) – (7.14), which ensure time feasibility and eliminate subtours. This set may be replaced by linear Constraints (7.29) – (7.35) below, where Z corresponds to a very large positive constant:

$$-w_{ik} + Z * x_{\mu_k ik} \leq Z - t_{\mu_k i} \qquad \forall k \in K, \forall i \in N \cup 0 \qquad (7.29)$$

$$-w_{s(u)k} + Z * x_{\mu_k s(u)k} \leq Z - t_{\mu_k s(u)} \qquad \forall k \in K, \forall u \in U \qquad (7.30)$$

$$w_{ik} - w_{jk} + Z * x_{ijk} \leq Z - s_i - t_{ij} \qquad \forall k \in K, \forall (i,j) \in \{(i,j): i \in N, j \in N \cup 0\} \qquad (7.31)$$

$$w_{ik} - w_{s(u)k} + Z * x_{is(u)k} \leq Z - s_i - t_{is(u)} \qquad \forall k \in K, \forall i \in N, \forall u \in U \qquad (7.32)$$

$$w_{s(u)k} - w_{f(u)k} + Z * x_{s(u)f(u)k} \leq Z - t_{s(u)f(u)} \qquad \forall k \in K, \forall u \in U \qquad (7.33)$$

$$w_{f(u)k} - w_{jk} + Z * x_{f(u)jk} \leq Z - t_{f(u)j} \qquad \forall k \in K, \forall j \in N \cup 0, \forall u \in U \qquad (7.34)$$

$$w_{f(u)k} - w_{s(\varphi)k} + Z * x_{f(u)s(\varphi)k} \leq Z - t_{f(u)s(\varphi)} \qquad \forall k \in K, \forall u \in T, \forall \varphi \in U \backslash \{u\} \qquad (7.35)$$

Constraints (7.18) ensure that the order is only unloaded at the designated customer. We linearize this constraint by replacing it with Inequalities (7.36) and (7.37):

$$(Z-1)z_h^{ki} - z_j^{ki} + Zx_{hjk} \leq 2Z - 2 \qquad \forall k \in K, \forall i \in N, \forall (h,j) \in A^{U} \text{ such that } h \neq i \qquad (7.36)$$

[10]

$$-z_h^{ki} + z_j^{ki} + Zx_{hjk} \leq Z \qquad \forall k \in K, \forall i \in N, \forall (h,j) \in A^U \text{ such that } h \neq i \qquad (7.37)$$

Constraints (7.19) and (7.20), which ensure that an order is unloaded or loaded at the designated location, are linearized using Inequalities (7.38) and (7.39), respectively.

$$-z_i^{ki} + z_j^{ki} + Zx_{ijk} \leq Z - 1 \qquad \forall k \in K, \forall i \in C, \forall j \in W \backslash (M \cup f(U)) \qquad (7.38)$$

$$-z_j^{ki} + z_i^{ki} + Zx_{ijk} \leq Z - 1 \qquad \forall k \in K, \forall i \in F, \forall j \in W \backslash (M \cup f(U)) \qquad (7.39)$$

Finally, we use Inequalities (7.40) and (7.41) below in order to linearize Constraints (7.21) and (7.22).

[10] $A^U = A \setminus \{(s(u), f(u) | u \in U)\}$

$$(7.21) \Rightarrow \quad \begin{aligned} & z_{s(u)}^{ki} + z_{f(u)}^{mi} = 2 \Rightarrow w_{f(u)m} \geq w_{s(u)k} \\ & \quad \Rightarrow w_{f(u)m} \geq w_{s(u)k} - Z\left(2 - z_{s(u)}^{ki} - z_{f(u)}^{mi}\right) \\ & \quad \Rightarrow w_{f(u)m} - w_{s(u)k} - Z * z_{s(u)}^{ki} - Z * z_{f(u)}^{mi} \geq -2 * Z \\ & \quad \Rightarrow \boldsymbol{-w_{f(u)m} + w_{s(u)k} + Z * z_{s(u)}^{ki} + Z * z_{f(u)}^{mi} \leq 2 * Z} \end{aligned} \quad (7.40)$$

$$(7.22) \Rightarrow \quad \begin{aligned} & z_{s(u)}^{ki} + z_{f(u)}^{mi} = 2 \Rightarrow w_{f(u)k} \geq w_{s(u)m} \\ & \quad \Rightarrow w_{f(u)k} \geq w_{s(u)m} - Z\left(2 - z_{s(u)}^{ki} - z_{f(u)}^{mi}\right) \\ & \quad \Rightarrow w_{f(u)k} - w_{s(u)m} - Z * z_{s(u)}^{ki} - Z * z_{f(u)}^{mi} \geq -2 * Z \\ & \quad \Rightarrow \boldsymbol{-w_{f(u)k} + w_{s(u)m} + Z * z_{s(u)}^{ki} + Z * z_{f(u)}^{mi} \leq 2 * Z} \end{aligned} \quad (7.41)$$

Based on the above, the final model may be solved by a commercial solver (CPLEX), and comprises objective function (7.1) and Constraints (7.2) – (7.7), (7.29) – (7.41), (7.15) – (7.16), (7.20) and (7.23) – (7.26).

7.4 Solution approach for DVRPMB-LT($\boldsymbol{\ell}$)

The model presented in Section 7.3.3 can be solved to optimality by a commercial solver (e.g. CPLEX). However, due to the large number of variables involved in the model, the solution may be limited only to cases of small size. For that reason, we have also developed a simple and straightforward heuristic procedure (framework) that is able to address cases of practical size. Specifically, we solve DVRPMB-LT(ℓ) using first a procedure that evaluates and identifies pairs of vehicles that may benefit from the transfer operation. Note that focusing on vehicle pairs is based on the assumption that each vehicle can participate in a single transfer along its route. Subsequently, we solve the problems of the promising pair of vehicles with an appropriate heuristic procedure. In the following, we introduce initially the solution framework for DVRPMB-LT(ℓ) (Section 7.4.1) and subsequently, in Section 7.4.2, we focus on the solution procedure for a single pair of vehicles.

7.4.1 The solution framework for DVRPMB-LT($\boldsymbol{\ell}$)

The proposed framework for solving the DVRPMB-LT(ℓ) is hereafter denoted as ***Load Transfer Algorithm, LTA***. The latter commences from the solution of the re-optimization problem that does not allow transfers. To do so, we employed the heuristic Branch-and-Price algorithm outlined in Chapter 4, which allows the incorporation of all DO to the vehicles *en route* or to vehicles stationed at depot. For convenience, we denote this algorithm as the ***No-***

Transfer Algorithm (NTA). Then, the solution framework comprises three additional steps as shown in Figure 7.7 and described below.

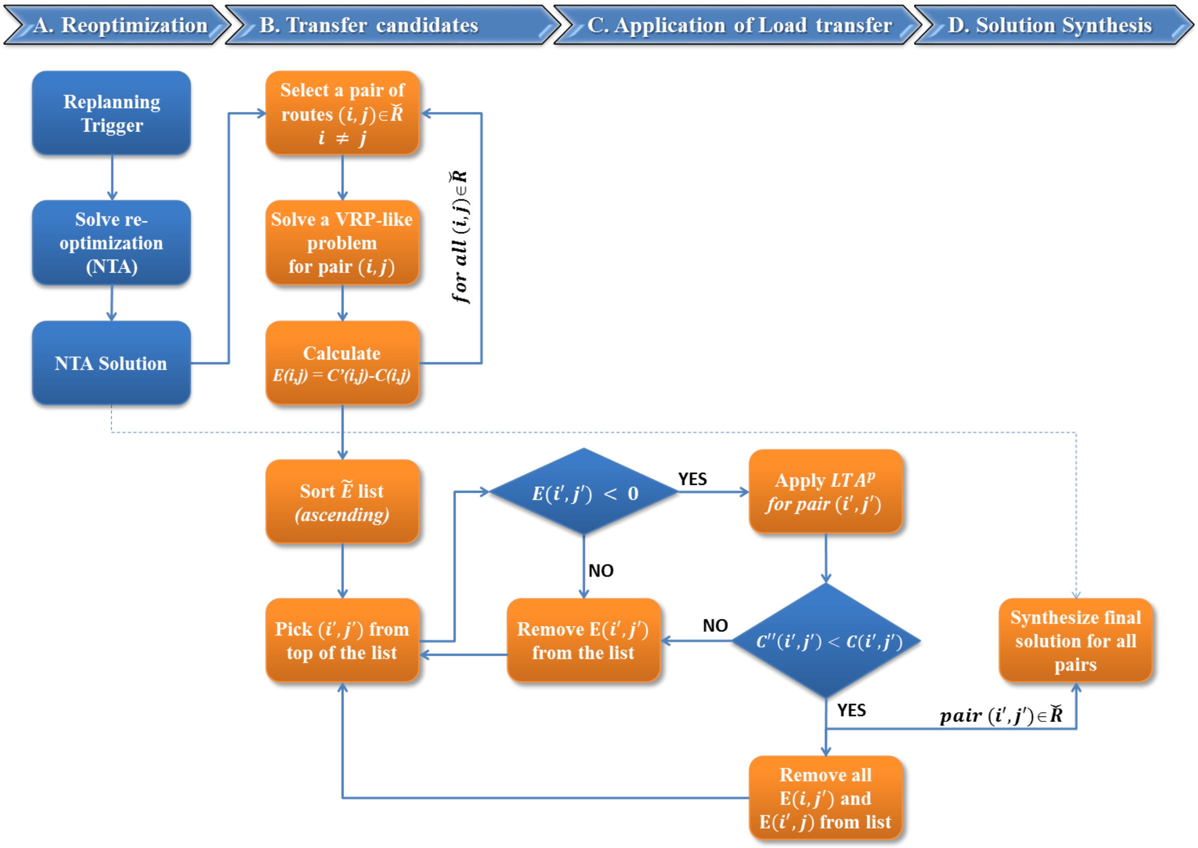


Figure 7.7. Solution framework for DVRPMB-LT(ℓ) (LTA)

In the second step of Figure 7.7 ("B. *Transfer candidates*") we identify candidate pairs of vehicle routes that may benefit from the transfer operation (typically, but not necessarily, vehicle routes with overlaps). To do so, for each pair of vehicle routes of the NTA solution (route set $\breve{R}$), we consider the related customers and solve a VRP-like problem with two vehicles. The routing cost of the resulting solution is then compared to the routing cost of the pair in the NTA solution. If the cost of the VRP-like problem is lower (even if the solution results to a single route), then this pair is a candidate to be further examined (for load transfer). In particular, for each pair of routes $(i, j), i \neq j$, let $E(i, j) = C'(i, j) - C(i, j)$, where $C'(i, j)$ is the cost corresponding to the VRP solution for the two vehicles and their respective customers, and $C(i, j)$ is the cost corresponding to the original re-optimization solution (NTA). In case $E(i, j)$ is negative, then pair (i, j) is a candidate for the transfer operation; if not, then this pair is discarded. Note that all possible vehicle pairs are evaluated at this step, even if a vehicle participates in more than one pairs. In general, each route will participate in $|\breve{R}| - 1$ pairs, and may correspond to up to $|\breve{R}| - 1$ negative $E(i, j)$ values. Let $\tilde{E}$ be the list containing the $E(i, j)$

values for all possible pairs. Before proceeding to step C, this list is sorted in ascending order, in order to evaluate first pairs with the higher negative value (and ensure that each vehicle will participate only in the best possible pair; see Step C below).

During the third step ("C. *Application of load transfer*"), we apply a heuristic algorithm, henceforth denoted as $\boldsymbol{LTA^p}$ and further described in Section 7.4.2, considering the pair $(i', j'), i \neq j$, with the lowest (negative) value in list $\tilde{E}$. If such pair does not exist, then the procedure terminates. If the solution for pair (i', j') is feasible and results to a routing cost, denoted as $C''(i', j')$, lower than the routing cost of NTA for this pair ($C(i', j')$), then the pair is qualified for the next step and all pairs (i, j') and (i', j) are removed from list $\tilde{E}$. If not, then this pair is discarded and the procedure iterates until list $\tilde{G}$ is empty (no remaining pairs for evaluation).

During the fourth step ("D. *Solution synthesis*"), we construct the final solution starting from the NTA solution and replacing the vehicle routes belonging to each pair of the third step with the solution of LTA for that pair.

7.4.2 Load transfer algorithm for a single pair of vehicles (LTA^p)

Assume that any order may be served by any vehicle of the pair under consideration. Consider the optimal (or near-optimal) solution of this VRP problem. If the delivery orders are served by the vehicles following the original assignment, then no transfer is required and the routing cost of the pair is optimal (or near optimal). If, however, one or more (delivery) orders are not served by the vehicle(s) according to the original assignment, then a transfer operation is needed. In this case, we identify the best transfer location by using an insertion-like algorithm and respecting all involved constraints. Finally, the solution obtained is further improved with post-optimization techniques. In particular, the method comprises three (3) distinct stages summarized below and further analyzed in the following Sections:
In particular, the method comprises three (3) distinct stages summarized below and is further analyzed in the following paragraphs:

Stage I. Routing: A VRP-like problem is solved by assuming all not yet served orders.
Stage II. Meeting: Identify the best available transfer location (if any).
Stage III. Post-Optimization: The resulting solution from Stage II is further improved.

7.4.2.1 Stage I: Routing

Consider a pair of vehicles each assigned with a set of not yet served orders (deliveries and/or pick-ups – DO). Each vehicle can either be located at a customer location or originate from the depot (new vehicle dispatched from depot under the NTA solution). At this stage, a VRP-like problem is solved by considering all unserved orders of the vehicle pair and the two available vehicles, without considering the original assignment of orders to vehicles. The network of the VRP-like problem ensures that: i) the first customer of each route will correspond to the current (starting) location of the vehicle, ii) no vehicle will travel to the starting location from any other customer, and iii) all involved times will be aligned to the re-optimization timestamp.

We solve the resulting VRP problem by using a Clark & Wright savings heuristic (Clark and Wright, 1964) followed by a Reactive Tabu Search metaheuristic (Osman and Wassan, 2002) as a post-optimization process. It should be noted here that early experimentation has indicated that the performance of the proposed load transfer framework (LTA) is highly dependent on the results of this Stage; thus, the selection of an appropriate VRP algorithm is important to the quality of the final solution.

7.4.2.2 Stage II: Identify best available transfer location

The solution obtained from Stage I provides the best possible assignment of customer orders to vehicles, without considering the original order assignments. We refer to the orders that are served in the solution of the VRP-like problem by a vehicle (or vehicles) other than the original one(s) as *transferred orders* (henceforth, denoted as *t-orders*). The *t-orders* (if they exist) need to be transferred prior to the service of the corresponding customers. To do so, we identify the most suitable location where the collaborating vehicles can potentially transfer the related loads. Note that it may not always be possible to identify such a location, since the process should respect the following constraints:

1. **Transfer constraints**: According to our original assumption (Section 7.3.2), each vehicle is restricted to be diverted to a transfer location at most once
2. **Precedence constraints**: Any *t-order* should exchange vehicles prior to serving the related customer
3. **Meeting constraints**: For the case of an *a priori* fixed transfer location, the vehicle that transfers the load to this location should arrive prior to the vehicle that receives the load, since for this case vehicles do not have to be simultaneously present at the transfer location. On the other hand, when the transfer takes place at a customer location, both vehicles need

to be present concurrently at that location. Thus, when a vehicle arrives to the transfer location prior to the other one, it has to wait until the other vehicle arrives.

4. **Load constraints**: The capacities of the collaborating vehicles should always be respected
5. **Time constraints**: The solution should respect all customer TW (if any), and both vehicles should return to the depot prior to the end of their available working horizon (T_{max}).

Using the VRP-like solution of the first stage, the potential transfer locations are considered and evaluated; i.e. each candidate transfer location of one route is temporarily inserted between two consecutive customers served by the other route. Each resulting route configuration is further improved by a post-optimization procedure. The pair of routes that incorporate the transfer location with the minimum cost is provided to the third phase of the method. In the remainder of this Section, we provide a formal description of the proposed heuristic.

Consider vehicles k_1 and k_2 and the sets of orders originally assigned to them $\Theta_{\mathrm{k}_1} = \{o_1, o_2, \dots, o_m\}$ and $\Theta_{\mathrm{k}_2} = \{o_{m+1}, o_{m+2}, \dots, o_{\ddot{m}}\}$, respectively. Assuming that each vehicle is currently located at positions μ_{k_1} and μ_{k_2}, respectively, and node 0 represents the depot, then each vehicle route resulting from Stage I can be represented as a vector, i.e. $R_{k_1} = [\acute{r}_{k_1}(\mu_{k_1}), \acute{r}_{k_1}(1), \dots, \acute{r}_{k_1}(m), \acute{r}_{k_1}(0)]$ and $R_{k_2} = [\acute{r}_{k_2}(\mu_{k_2}), \acute{r}_{k_2}(1), \dots, \acute{r}_{k_2}(\ddot{m}), \acute{r}_{k_2}(0)]$, where $\acute{r}_k(i)$ represents the sequence of order $i \in \Theta_k, k \in \{k_1, k_2\}$ in vehicle route $R_k, k \in \{k_1, k_2\}$. We also denote as $\mathrm{r_k(e_k)}, \mathrm{k} \in \{1,2\}$ the first node in the route of vehicle k that corresponds to a *t-order*, where $e_k, k \in \{k_1, k_2\}$ corresponds to the sequence of this node in the vehicle route. The e_k nodes are crucial for ensuring the precedence constraints.

In the following, we describe separately the procedure for the two transfer cases.

<u>Transfer operation at a customer location</u>

Figure 7.8 provides an overview of the process used to identify the best transfer location among the candidate locations of the not yet served customers. The related algorithm operates in an iterative manner. During each iteration of the process, a node j of route R_{k_1} is inserted between two consecutive nodes of route R_{k_2}, where $k_1, k_2 \in \{1,2\}, k_1 \neq k_2$.

In order to impose precedence constraints, this process investigates only the nodes of each route sequence that are prior to nodes $\mathrm{r_k(e_k)}, \mathrm{k} \in \{1,2\}$. In case the insertion of node j to route $R_{k'}$ maintains feasibility (satisfying the time and capacity constraints), and the total cost (distance) of the updated routes is the lowest one so far, location $r_{k'}(j)$ is selected as the transfer location (U^*) and the corresponding pair of routes is kept as the best found to this point (R^*) with corresponding cost C^*. Prior to identifying a solution that improves the total routing cost, we also apply a post-optimization procedure (Line 11 of Algorithm 1) similar to the one described in the third stage of the process (see below). It is important to apply this refinement to every solution (even to the infeasible ones), since an infeasible one may be rendered feasible (due to the possible re-arrangement of the customers involved in the routes).

Algorithm 1: Identify best transfer location

Input: $\{R_1, R_2\}$, C^0 // C^0 the total routing cost resulted from NTA
Output: R^*, C^*, T^*
1 $C^* = C^0$, $R^* = \{R_1, R_2\}$ // Initialization
2 **For** each $k_1 \in \{1,2\}$ // FOR LOOP A
3 **For** each $k_2 \in \{1,2\}, k_2 \neq k_1$ // FOR LOOP B
4 **If** $k_1 = 1$ **then** $a = m$ **else** $a = \ddot{m} - m$ **end**
5 **For** each $j = 1$ to e_{k_2} **do** // FOR LOOP C
6 **For** each $i = 1$ to $e_{k_1} - 1$ **do** // FOR LOOP D
7 $R_{k_1} = [\acute{r}_{k_1}(\mu), \acute{r}_{k_1}(1), \dots, \acute{r}_{k_1}(i), \acute{r}_{k_2}(j), \acute{r}_{k_1}(i+1), \dots, \acute{r}_{k_1}(0)]$
8 $R_{k_2} = [\acute{r}_{k_2}(\mu), \acute{r}_{k_2}(1), \dots, \acute{r}_{k_2}(a), \acute{r}_{k_2}(0)]$
9 Compute cost $\ddot{C} = C^{ij} = C(r_1) + C(r_2)$
10 **If** $\ddot{C} < C^*$
11 $(C^*, R^*) = \boldsymbol{Opt}(C^*, R^*)$ // Solution improvement
12 **If** R_{k_1} and R_{k_2} feasible (Time & Load constraints)
13 $C^* = \ddot{C}$
14 $R^* = \{R_{k_1}, R_{k_2}\}$
15 $U^* = \acute{r}_{k_2}(j)$
16 **End If**
17 **End if**
18 **End for loop** // FOR LOOP D
19 **End for loop** // FOR LOOP C
20 **End for loop** // FOR LOOP B
21 **End for loop** // FOR LOOP A

Figure 7.8. Pseudo-code for identifying the most appropriate transfer location among the locations of the customers not yet served

Fixed transfer location

In this case, the procedure attempts to identify the best time instance for the vehicles to visit the fixed transfer location (U_f). At each iteration, the procedure evaluates visiting the transfer location between two consecutive nodes of route R_1 or R_2. Of course, the operation in this case is also performed only for the nodes of the two routes that are prior to $\acute{r}_k(e_k), k \in \{k_1, k_2\}$. A valid inclusion of U_f in each route satisfies time and load constraints. An additional feasibility criterion is the sequence of visiting the transfer location (meeting constraint); the vehicle discharging the load should arrive at U_f prior to the vehicle receiving it. In case both vehicles discharge loads, they have to be at the transfer location at the same time. If feasibility is maintained and the total cost is the best found to this point, then the fixed location is used for the transfer (U^*), otherwise it is discarded and the solution resulted from the solution of NTA is used. The algorithm is outlined in Figure 7.10.

Algorithm 2: Incorporation of fixed transfer location

Input: $\{R_1, R_2\}, C^0, U_f$ // C^0 the total routing cost resulted from NTA
Output: R^*, C^*

$C^* = C^0, R^* = \{R_1, R_2\}$ // Initialization
For each $j \in \{1, \ldots, e_2\}$ **do** // FOR LOOP A
  For each $i \in \{1, \ldots, e_1\text{-}1\}$ // FOR LOOP B
    $R_{k_1} = [\acute{r}_1(\mu), \acute{r}_1(1), \ldots, \acute{r}_1(i), U_f, \acute{r}_1(i+1), \ldots, \acute{r}_1(0)]$
    $R_{k_2} = [\acute{r}_2(\mu), \acute{r}_2(1), \ldots, \acute{r}_2(j), U_f, \acute{r}_2(j+1), \ldots, \acute{r}_2(0)]$
    Compute cost $\ddot{C} = C^{ij} = C(r_1) + C(r_2)$
    If $\ddot{C} < C^*$
      If R_{k_1} and R_{k_2} feasible (Time & Load constraints)
        $C^* = \ddot{C}$
        $R^* = \{R_{k_1}, R_{k_2}\}$
      End If
    End if
  End for loop // FOR LOOP B
End for loop // FOR LOOP A

Figure 7.9. Pseudo-code of heuristic approach for incorporating the fixed transfer location

7.4.2.3 Stage III: Post-optimization

After identifying the best route sequence, including the location for the load transfer operation, a simple post optimization procedure attempts to refine the solution by node interchanging moves (2-opt) (Croes, 1958; Lin, 1965) are employed a) within any single route and b) between the routes of the pair as illustrated in Figure 7.10. Appropriate checks are also conducted to ensure that no constraints are violated.

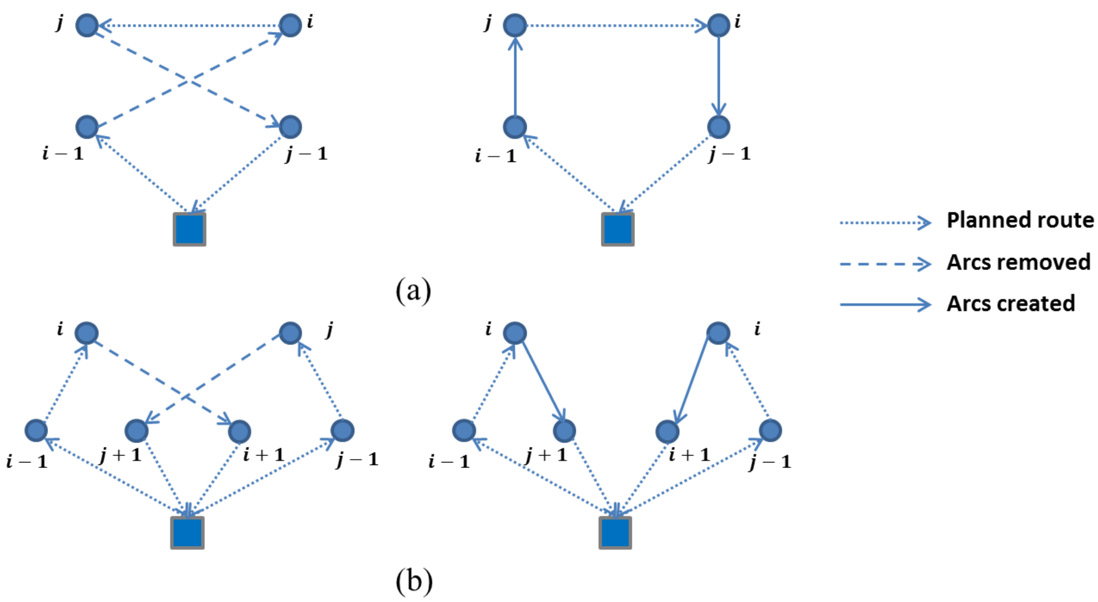


Figure 7.10. Interchange moves; (a) within a single route, (b) between a route pair

7.5 Computational experiments

To assess the benefits of load transfers within the DVRPMB setting, we compare the solutions provided allowing load transfers to the ones that do not allow transfers. The experimental investigation is structured as follows: in Section 7.5.1 we evaluate the performance of the proposed heuristic approach (LTA) for solving DVRPMB-LT(ℓ) for a pair of vehicles with respect to its optimal counterpart. In Section 7.5.2 we investigate the performance of LTA compared to NTA by considering pairs of (overlapping) vehicle routes in order to assess the advantage of allowing transfers in the re-optimization problem. To do so, we consider two typical operating scenarios; i) both vehicles of the investigated pair are *en route* at the re-optimization timestamp (Section 7.5.2.1), and ii) one vehicle is *en route* and the other is located at the depot (Section 7.5.2.2). In Section 7.5.3 we evaluate the benefits of the proposed framework for the solution of DVRPMB-LT(ℓ) where more than two vehicles are involved (LTA). Finally, in Section 7.5.4 we investigate the performance of load transfers under different re-optimization strategies for the entire dynamic problem (DVPRMB-LT).

The experiments presented below were conducted using a Quad-Core Intel i7 processor of 2.8GHz and 4GB of RAM. The MILP model was solved using the commercial MILP solver TOMLAB/CPLEX Version 7.5 (R7.5.0). The solver's default settings were used, and the initialization computational time was not included in the computational time values presented below.

7.5.1 Assessment of the LTA^p heuristic

In order to assess the performance of the proposed LTA^p heuristic, we considered re-optimization problems involving a single pair of vehicles *en route*. At the re-optimization timestamp, each vehicle is assigned with a set of static (delivery) orders not yet served, where new DO have been received and need to be incorporated in the current plan. We then solve the underlying problems with LTA^p and compare the solution with its optimal counterpart. In order to ensure that the solution of the re-optimization problem will require only two vehicles, we assumed no limitations w.r.t. time windows, shift duration and capacity.

For this experimental study, we have employed randomly generated data within a service area of $1 \times 1\ km^2$. We generated test instances varying: a) the number of static (delivery) orders per vehicle ($C_k, k \in \{1,2\}$), and b) the number of DO (F). The number of either type of orders was varied from 2 to 7 for each set, leading to a total of 36 test instances (see Table 7.3). The total number of nodes per instance depends on the number of static and dynamic orders considered; for example, if $|C_k| = 4$ and $|F| = 4$, the total number of nodes ($|N|$) in the network for this test instance is 15 (8 static delivery nodes, 4 DO nodes, 2 nodes for the vehicle starting locations, and one for the depot). For each test instance, we generated 10 different problems by assigning the node locations randomly within the defined area using a uniform distribution. Thus, the full problem set involves 360 test problems. For each test problem, delivery orders were randomly assigned to each vehicle, and the resulting route (from the current location to the depot) was improved by a 2-opt procedure (Li, 1965). Finally, we assumed that one distance unit equals to one time unit.

Table 7.3. Parameters of test problems

Parameter	Description	Values (levels)	# of levels
C_k	Deliveries per vehicle	2, …, 7	6
F	Pickup orders (DO)	2, …, 7	6
$\wp$	Test problems/instance	1,…,10	10

Each generated test problem was originally solved using the B&P heuristic method of Chapter 4.7 in order to optimally assign DO to vehicles *en route,* without transfers (NTA). Subsequently, each test problem was solved by allowing load transfers: i) to optimality, by solving the MILP model of Section 7.3.3 with a commercial MILP solver, and ii) using LTA^p. For the load transfer operation, we investigated both the case of fixed transfer location and the case of load transfers at the locations of any not yet served customers. The fixed transfer location was selected to be the center of mass of the customer nodes. Table 7.4 summarizes the algorithms studied in this experimental phase and the corresponding designations to be used hereafter.

Table 7.4. Description of the algorithms employed

Alias	Description
NTA	Solution with no transfers
LTA_f^{opt}	Solution of LTA^p considering a fixed transfer location and using the MILP solver
LTA_d^{opt}	Solution of LTA^p considering load transfer at the location of any not yet served customer and using the MILP solver
LTA_f	Solution of LTA^p considering a fixed transfer location and using the heuristic
LTA_d	Solution of LTA^p considering load transfer at the location of any not yet served customer and using the heuristic approach

Table 7.5 summarizes the results obtained as an average of all (10) test problems ($|\wp|$) per instance. The first three columns of the table denote the number of delivery orders per vehicle, the number of DO, and the total number of nodes, respectively. The remaining columns illustrate for each one of the investigated algorithms the total distance traveled (TD), the number of test problems (out of 10) that a transfer operation took place ($|U|$) and the computational time in seconds (CT). Finally, the bottom section of the Table presents the average values per column ('*Mean*') and the percentage improvement ('*%Imp*') w.r.t the reference case (i.e. NTA solution). Note that results are not provided for LTA_d^{opt} for $|C_k| \in \{6,7\}$, since LTA_d^{opt} could not solve the related cases within reasonable computational times (due to the problem size).

Table 7.5. Detailed results for NTA and the investigated LTA algorithms

$\|C_k\|$	$\|F\|$	$\|N\|$	NTA		LTA_f^{opt}			LTA_f			LTA_d^{opt}			LTA_d		
			TD	**CT**	**TD**	**|U|**	**CT**	**TD**	**|U|**	**CT**	**TD**	**|U|**	**CT**	**TD**	**|U|**	**CT**
2	**2**	9	3,218.6	0.1	3,032.2	7	0.3	3037.4	7	1.8	2,878.0	9	105.7	2,882.7	8	0.3
	3	10	3,425.4	0.1	3,340.3	8	0.5	3382.5	6	1.4	3,038.5	9	246.8	3,067.6	9	0.7
	4	11	3,435.5	0.2	3,297.6	6	3.6	3297.6	6	1.8	3,162.3	7	3,486.3	3,194.3	7	0.7
	5	12	3,608.2	0.5	3,517.2	7	2.7	3608.2	5	1.5	3,293.0	9	3,600.2	3,301.6	9	1.7
	6	13	3,651.2	1.3	3,527.0	8	25.1	3560.7	6	1.7	3,294.1	9	3,229.0	3,321.5	9	3.1
	7	14	3,825.9	2.4	3,694.4	8	19.0	3768.6	7	3.8	3,537.0	9	3,364.6	3,537.7	9	3.6
3	**2**	11	3,770.2	0.1	3,494.9	9	3.7	3578.8	8	2.5	3,223.1	9	3,208.8	3,299.0	8	1.2
	3	12	4,037.4	0.2	3,760.7	9	8.7	3852.1	7	1.5	3,556.3	10	3,600.2	3,603.4	8	1.3
	4	13	3,912.0	0.6	3,592.9	8	13.1	3730.1	6	2.7	3,192.0	8	3,631.7	3,244.3	8	2.7
	5	14	4,196.7	1.2	3,981.5	8	27.7	4098.0	7	4.1	3,912.3	9	3,601.2	3,937.7	7	3.5
	6	15	4,025.2	3.5	3,613.8	9	290.7	3657.9	8	11	3,297.6	10	3,218.7	3,317.8	10	4.8
	7	16	4,037.3	5.5	3,895.4	8	204.7	3906.9	8	11.8	3,653.7	10	5,249.4	3,684.4	10	5.3
4	**2**	13	4,204.5	0.2	3,877.3	9	2.0	3900.1	9	2	3,556.6	10	3,600.4	3,590.0	10	3.1
	3	14	4,269.5	0.6	3,779.2	10	4.0	3806.3	10	4	3,220.5	10	3,602.5	3,260.2	10	4.3

$\lvert C_k \rvert$	$\lvert F \rvert$	$\lvert N \rvert$	NTA		LTA_f^{opt}			LTA_f			LTA_d^{opt}			LTA_d		
			TD	**CT**	**TD**	**|U|**	**CT**	**TD**	**|U|**	**CT**	**TD**	**|U|**	**CT**	**TD**	**|U|**	**CT**
	4	15	4,255.3	1.4	4,147.5	7	5.5	4153.2	7	5.5	3,677.0	10	3,605.5	3,717.0	10	9.1
	5	16	4,540.7	3.8	4,150.5	9	9.8	4263.6	9	9.8	3,912.7	10	3,656.1	3,932.7	10	9.8
	6	17	4,528.2	8.1	3,917.9	10	52.7	4006.5	10	11.2	3,475.6	10	4,261.6	3,546.5	10	12.6
	7	18	4,510.9	16.8	4,043.3	10	593.4	4161.6	10	11.6	3,587.5	10	4,567.4	3,709.9	10	14
5	**2**	15	4,151.0	0.7	3,636.1	10	67.1	3729.5	10	8.1	3,301.1	10	3,813.3	3,338.2	10	7
	3	16	4,987.0	0.5	4,165.3	9	59.6	4262.9	9	7.6	3,724.1	10	3,794.9	3,725.0	10	5.9
	4	17	4,512.9	0.7	3,932.0	10	408.7	4034.9	10	6.3	3,708.9	10	4,974.2	3,758.7	10	8.4
	5	18	4,920.9	1.0	4,459.3	9	1551.2	4559.6	8	8.9	3,910.7	10	5,558.3	4,018.0	10	9.6
	6	19	4,817.8	1.8	4,503.4	10	1271.1	4591.2	8	12	3,995.3	10	7,236.4	4,154.8	10	12.7
	7	20	5,140.5	3.4	4,458.5	10	1142.7	4647.3	9	13.5	4,121.5	10	8,013.2	4,126.3	10	12.8
6	**2**	17	4,589.0	0.5	4,115.2	10	189.9	4191.6	10	8	-	-	-	3,764.9	10	6.9
	3	18	4,972.9	0.5	4,355.3	9	183.5	4506.6	9	8.1	-	-	-	3,990.1	9	6.2
	4	19	4,816.6	0.9	4,137.2	10	1001.6	4366.4	10	9.7	-	-	-	3,914.8	10	9.7
	5	20	5,206.0	1.5	4,428.6	10	2301.1	4539.8	10	9.4	-	-	-	3,983.9	10	10.7
	6	21	5,370.7	3.0	4,529.1	9	1078.3	4534.9	9	10.5	-	-	-	4,234.6	10	11.6
	7	22	5,401.2	3.9	4,851.4	10	1652.4	4874.8	10	12.8	-	-	-	4,215.7	10	12.2
7	**2**	19	5,337.7	0.5	4,341.5	10	293.0	4439.7	10	8.4	-	-	-	3,961.0	10	9.4
	3	20	5,128.0	1.1	4,358.4	9	2529.7	4475.8	9	8.3	-	-	-	3,915.8	10	10.6
	4	21	5,322.3	1.4	4,451.3	10	1843.0	4607.8	9	10.9	-	-	-	4,086.9	10	11.9
	5	22	5,159.7	2.8	4,454.8	10	1056.6	4651.6	9	10.6	-	-	-	4,235.0	10	10.4
	6	23	5,539.8	4.3	4,952.4	8	812.1	4972.1	8	12.3	-	-	-	4,406.9	10	12.8
	7	24	5,350.9	6.6	4,806.6	8	3734.0	4878.7	8	15.8	-	-	-	4,406.6	10	23
	Mean		***4,504.9***	*2.3*	***4,044.4***	*8.9*	*623.4*	***4,128.8***	*8.4*	*7.5*	***3,509.6***	*9.5*	*3,884.4*	***3,732.9***	*9.5*	*7.6*
	%Imp				***10.2%***			***8.3%***			***22.1%***			***17.1%***		

In the following, we analyze the performance of the heuristic approach compared to its optimal counterpart. The performance of LTA compared to NTA is analyzed in Section 7.5.2. Figure 7.11 presents the average deviation of $\mathrm{LTA_f}$ and $\mathrm{LTA_d}$ from their optimal counterparts $\mathrm{LTA_f^{opt}}$ and $\mathrm{LTA_d^{opt}}$, respectively. Performance is assessed in terms of routing costs, and results are presented for each number of deliveries per vehicle (C_k) averaged over all DO levels and test problems ($|\wp|$). According to this figure, the proposed heuristic seems to be highly competitive with respect to the optimal solutions of the MILP solver. In particular, for the fixed transfer location case, the solutions obtained by the heuristic have an average deviation of 1.8% from the optimal ones. When load transfer is allowed at any not yet served customer, the average deviation of the heuristic solution from the optimal is about 1.0%. The deviation for both cases seems to be consistent throughout the different numbers of delivery orders per vehicle (value C_k).

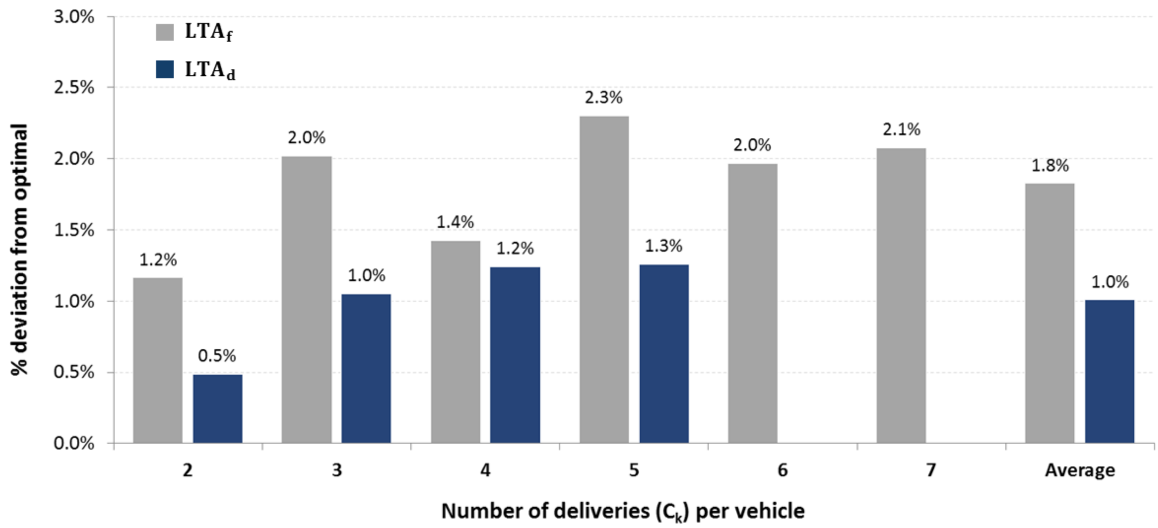


Figure 7.11. Overall heuristic assessment (deviation from optimal)

Regarding computational effort, Table 7.5 shows that the proposed heuristic arrives at the solution in less than 10 seconds on average for all cases. This validates its efficiency and suitability for practical dynamic applications for which fast solutions are required. It should be noted here that the greatest portion of computational effort is spent in solving the VRP-like problem (Stage I of the algorithm described in Section 7.4.2.1).

7.5.2 Re-optimization with load transfers for a pair of vehicle routes

In this Section we consider again the case of two vehicles, and investigate further the benefit of load transfers during re-optimization w.r.t. the policy that does not allow such transfers. To do so, we investigate the application of load transfers (LTA) for a pair of vehicles with overlapping routes and compare it with the solution provided by NTA. Recall that LTA is applied after the solution of the re-optimization problem, which typically results to overlapping routes, as described in Section 7.3.1. We investigate two operating scenarios: a) a case in which both vehicles are *en route*, and b) a case in which one of the two vehicles is located at the depot (i.e. dispatched from depot to serve newly received DO after the solution of NTA, but before the application of LTA).

7.5.2.1 Both vehicles *en route*

This is the case of Section 7.5.1. Thus, no new experiments are conducted, but the results presented already in Table 7.5 are further analyzed.

Figure 7.12 illustrates the performance of LTA_f and LTA_d (heuristic procedure of Section 7.4.2) in terms of percentage difference (improvement) from the NTA solution (B&P heuristic of

Chapter 4.7), that is $\%(TD_{NTA} - TD_{LTA})/TD_{NTA}$. The results are presented per number of deliveries per vehicle (C_k). For each C_k value the Figure presents the average cost improvements over all DO levels and test problems. The overall average performance is also provided.

According to the Figure, both LTA algorithms outperform NTA in all investigated instances in terms of routing costs. In particular, $\mathrm{LTA_f}$ and $\mathrm{LTA_d}$ offer savings of 7.9% and 16.5% on average, respectively. As expected, the option of allowing loads to be transferred at the location of any not yet served customer ($\mathrm{LTA_d}$) leads to significantly higher savings. Furthermore, the performance of both LTA algorithms improves w.r.t. the number of delivery orders per vehicle, as expected, since the longer the routes, the more chances for significant overlaps and more possibilities for load transfers. This indication leads also to the assumption that load transfer policies might be more preferable during early re-optimization cycles, when vehicles have not executed significant portion of their routes.

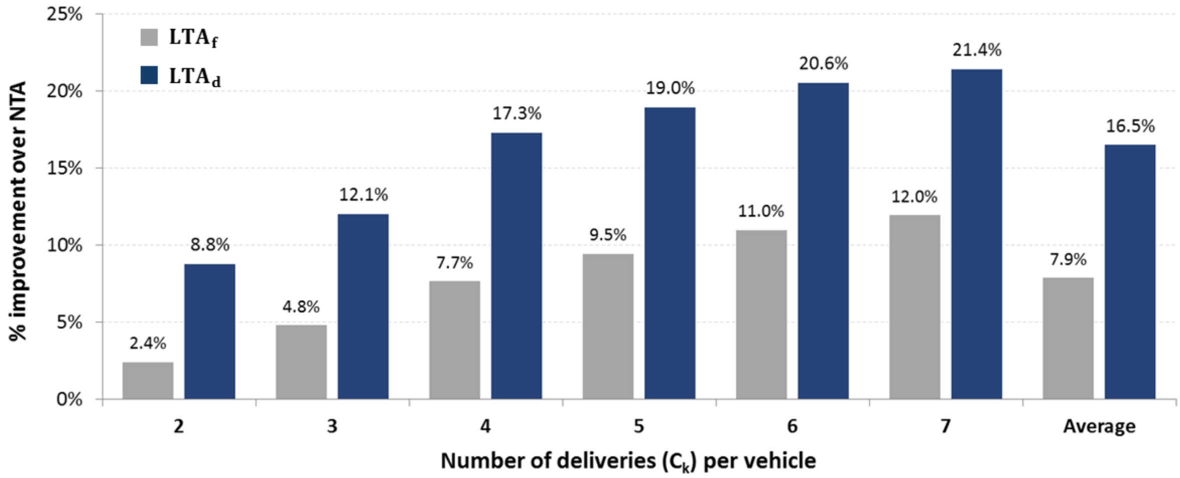


Figure 7.12. Average performance of LTA w.r.t. the number of delivery orders per vehicle

7.5.2.2 The case of one vehicle located at the depot

The tests for this case were generated based on the customer coordinates of the Solomon benchmarks (Solomon, 1987; see description on benchmarks in Chapter 5, Section 5.4.1.2). In order to assess the impact of customer geographical distribution on the performance of LTA, we used Solomon instances from both the R and C configurations. For each configuration, we generated cases consisting of 15, 25, and 50 customer orders. The number of customers was limited to 50, since only two vehicles were involved. For each of those 6 test sets (2 types of geographical distribution, 3 levels of number of customers), we investigated cases with and without TW, resulting in a total of 12 test cases, as shown in Table 7.6.

For the test cases with 15 customers and TW, we generated one test problem for every benchmark instance in the R1 and C1 datasets, i.e. 11 and 8 test problems, respectively (note that R101 and C101 instances were excluded because their tight TW profile cannot offer any savings by applying load transfers). For the test cases with 25 or 50 customers and TW, we generated one test problem for every instance in the R2 and C2 datasets (i.e. 11 and 8 test problems, respectively) for each level of the number of customers. It is noted that the R2 and C2 datasets permit the assignment of 25 and 50 customers in 2 vehicles. For the test cases (of 15, 25 and 50 customers) with no TW, we employed instances vrpnc8 and vrpnc14 of Christofides *et al.* (1979). These instances do not have customer TW but use the same customer coordinates as in Solomon's R1 and C1 datasets. For each one of the vrpnc8 and vrpnc14, we generated 10 different test problems (using random selection of customers from the original vrpnc8 and vrpnc14 instances). Based on the above, a total of 117 test problems were generated, as shown in Table 7.6.

Table 7.6. Test cases

Test case	Corresponding Benchmark	Customer Orders (V)	Geographical Distribution (R^2)	Time windows (TW)	# Test problems
1	R1	15	R	TW	11
2	vrpnc8	15	R	NoTW	10
3	C1	15	C	TW	8
4	vrpnc14	15	C	NoTW	10
5	R2	25	R	TW	11
6	vrpnc8	25	R	NoTW	10
7	C2	25	C	TW	8
8	vrpnc14	25	C	NoTW	10
9	R2	50	R	TW	11
10	vrpnc8	50	R	NoTW	10
11	C2	50	C	TW	8
12	vrpnc14	50	C	NoTW	10

For each test problem, customers were randomly selected from their original corresponding benchmark problem. Note that we skewed the selection towards consecutive customers for the C configuration instances (due to the sequential order of customers within clusters in the original benchmark instances). Customer characteristics, shift duration and capacity restrictions were considered as per the original benchmarks.

The following also hold for all test problems: i) for each test problem, we randomly selected static (delivery) orders and DO as per Table 7.7; ii) the initial solution corresponds to the optimal assignment of all delivery orders to one of the two available vehicles, while the other is located at the depot, iii) re-optimization is triggered at the time the vehicle arrives at

$(\breve{n}+1)-th$ customer (see Table 7.7); iv) as in previous Sections, for each test problem, we applied NTA, $\mathrm{LTA_f}$ and $\mathrm{LTA_d}$; v) the fixed transfer location was considered to be the centre of mass of the customer nodes.

Table 7.7. Customer characteristics of the generated instances

Number of customer orders			$\tilde{n}$
Total	**Deliveries**	**DO**	
15	9	6	2
25	13	12	2
50	32	18	5

Table 7.8 summarizes the results obtained for each test case. Results have been averaged over all test problems within a test case. The first three columns of the Table indicate the characteristics of each case, according to Table 7.6. The subsequent column sets present the performance of NTA, $\mathrm{LTA_f}$ and $\mathrm{LTA_d}$; i.e., the total distance travelled (TD), the total computational effort (CT), the number problems in which a transfer operation took place (|U|) and the percentage deviation of each LTA strategy from NTA (%Dev).

Table 7.8. Detailed results for NTA and the proposed LTA algorithms

Test case	V	R^2	TW	# Insta -nces	NTA		$\mathrm{LTA_f}$				$\mathrm{LTA_d}$			
					TD	CT	TD	|U|	CT	%Dev	TD	|U|	CT	%Dev
1	**15**	**R**	**TW**	11	297.1	3.1	281.2	7	4.6	**5.4%**	267.1	10	4.7	**10.1%**
2	**15**	**R**	**NoTW**	10	287.7	6.3	257.6	9	4.3	**10.5%**	249.4	10	6.1	**13.3%**
3	**15**	**C**	**TW**	8	223.7	3.4	176.0	7	4.1	**21.3%**	168.7	8	5.9	**24.6%**
4	**15**	**C**	**NoTW**	10	210.5	7.8	183.9	9	4.5	**12.6%**	180.3	10	6.3	**14.3%**
5	**25**	**R**	**TW**	11	469.4	6.1	402.6	9	25.1	**14.2%**	397.8	11	27.2	**15.2%**
6	**25**	**R**	**NoTW**	10	386.6	18.2	335.0	10	21.7	**13.4%**	325.1	10	22.6	**15.9%**
7	**25**	**C**	**TW**	8	285.2	5.4	253.9	8	20.3	**11.0%**	236.3	8	13.9	**17.1%**
8	**25**	**C**	**NoTW**	10	304.2	30.4	260.0	10	19.3	**14.5%**	251.4	10	16.3	**17.4%**
9	**50**	**R**	**TW**	11	688.0	15.1	620.6	11	43.8	**9.8%**	598.8	11	41.6	**13.0%**
10	**50**	**R**	**NoTW**	10	571.4	33.9	503.0	10	42.6	**12.0%**	496.6	10	40.3	**13.1%**
11	**50**	**C**	**TW**	8	547.5	15.2	475.4	8	41.9	**13.2%**	440.7	8	49.4	**19.5%**
12	**50**	**C**	**NoTW**	10	405.8	45.1	359.9	10	43.1	**11.3%**	341.6	10	45.9	**15.8%**
				Mean	***389.8***	***15.8***	***342.4***	***9.0***	***22.9***	***12.4%***	***329.5***	***9.7***	***23.4***	***15.8%***

The table clearly shows the superiority of both $\mathrm{LTA_f}$ and $\mathrm{LTA_d}$ over NTA for all investigated cases. Another interesting observation is that load exchanges are performed in more than 90% of the investigated cases (value $|U|$), illustrating the significance of employing a transfer policy in such a setting. LTA provides the solution in less than 1 minute (even for the case of 50 customers).

Figure 7.13 illustrates the overall performance of LTA w.r.t. geographical distribution and TW parameters. The results shown are the averages of all related test problems. The performance is reported as a percentage improvement (saving) over NTA. According to the Figure, LTA outperforms NTA in all cases with a tendency of savings to increase when customers are clustered (C configuration). This can be attributed to the fact that the vehicle *en route* (assigned with delivery orders) may travel to different clusters. When the DO are introduced, under NTA both vehicles may be forced to visit the same clusters, leading to inferior results.

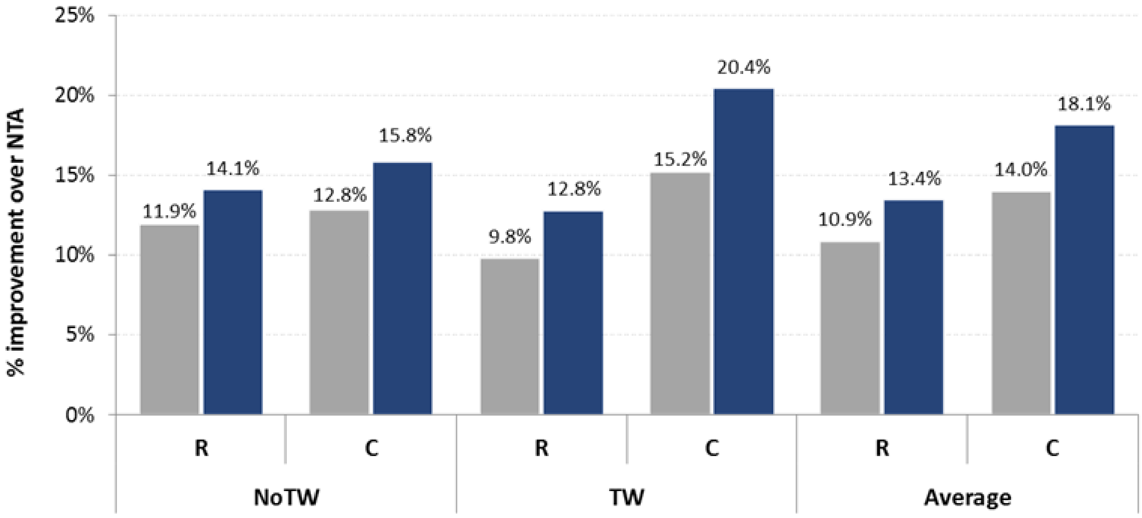


Figure 7.13. Average performance of LTA w.r.t. geographical distribution and TW patterns

According to Figure 7.13, LTA_d consistently outperforms LTA_f in all cases and this superior performance seems to be enhanced in clustered cases. What is interesting to note in this operating scenario and in contrast to the case investigated in Section 7.5.2.1 (both vehicles *en route*), is that LTA_f seems to be more competitive to LTA_d, especially for cases where customers are distributed uniformly (R) and TW are not present. This may be caused by the flexibility of the vehicle located at the depot to travel directly to the fixed transfer location and pick-up the transferred loads. Finally, LTA seems to offer higher savings when TW are imposed and customers are clustered, compared to the non-TW cases. This may be attributed to the fact that TW may force vehicles under NTA to re-visit the same clusters more than once, which will cause higher costs due to the typically long inter-cluster distances. This effect may be moderated when load transfers are introduced (i.e. each vehicle travels to a single cluster).

7.5.3 Re-optimization with load transfers for multiple (more than two) vehicles

To investigate the performance of LTA in cases in which more than two vehicles are involved, we employed indicative cases with and without TW. For the former, we used two (2) of the benchmark instances of Solomon (1987), i.e. R109 and R112 with average TW width of 25% and 50% of the allowed working time (T_{max}), respectively. Note that computational experiments not presented in this Section have illustrated that there is very limited benefit from allowing transfer operations in cases in which the average TW width is relatively tight (i.e. less than 25% of T_{max}). In order to investigate cases with no TW, we employed the *vrpnc8* instance of Christofides *et al.* (1979) that uses the same customer coordinates as the Solomon R109 and R112 instances.

For each one of the three instances, we generated 5 different problems (different selection of delivery orders), resulting in a total of 15 test problems. 50% of delivery orders were randomly selected from the 100-customer problem; for the remaining 50 orders (of the 100-customer problem), we randomly assigned a time of arrival ($h_i, i \in F$) during the window $[0{,}0.75 * T_{max}]$ according to a continuous uniform distribution. We selected 33% of those that arrive earlier to form the set of DO.

The following also apply in the current experimental cycle: (i) the initial solutions (assignment of delivery orders to routes) were obtained by a Clark & Wright savings heuristic (Clark and Wright, 1964) followed by a Reactive Tabu Search metaheuristic (Osman and Wassan, 2002) used as post-optimization; (ii) re-optimization is triggered at the time when the last DO have been received; (iii) the re-optimization problem was solved following the framework presented in Section 7.4.1 (Figure 7.7); (iv) for the following investigation, we employed only the $\mathrm{LTA_d}$ algorithm.

Table 7.9 provides relevant information per instance as an average of all test problems of this instance; in particular, we present the total number of orders considered during re-optimization (Total), the total number of delivery orders (SO) and the number of DO. Additionally, information regarding the number of SO and DO per route is provided. The number of routes reported corresponds to the solution prior to LTA.

Table 7.9. Information of each investigated instance during the re-optimization cycle

Instance	Total	SO	DO	# Routes	SO/route	DO/route
R109	46.5	29.5	17	8.5	3.5	2.0
R112	48.6	31.6	17	7.2	4.4	2.4
vrpnc8	48.6	31.6	17	7.6	4.2	2.2

The results obtained are summarized in Table 7.10 per instance, averaged over all test problems. The first two columns correspond to the instance and the percentage of routes involved in load-transfers w.r.t. the total number of routes after the application of NTA ($\%\hat{P}$). The subsequent columns are grouped in two sets: a) the first set (4 columns) presents the performance of LTA vs. NTA w.r.t. the complete solution (all routes), and b) the second set presents the performance of LTA vs. NTA w.r.t. the routes participating in load-transfer operations (*transfer pairs*). In each set, we report the routing cost obtained from NTA and $\mathrm{LTA_d}$, the percentage savings of $\mathrm{LTA_d}$ over NTA (*%Dev*) and the total computational effort (CT) in seconds. The CT for the first set (complete solution) comprises the time for the solution of the re-optimization problem and the application of LTA; CT for the second set reports only the average time for the solution of LTA involving the pairs for which transfers took place. Finally, the last row of the Table reports average performance indicators per instance.

Table 7.10. Detailed results for NTA and $\mathrm{LTA_d}$

Instance	$\%\hat{P}$	Complete solution				Transfer pairs			
		NTA	**$\mathrm{LTA_d}$**	**%Dev**	**CT**	**NTA**	**$\mathrm{LTA_d}$**	**%Dev**	**CT**
R109	16.0%	762.7	751.5	*1.5%*	15.3	196.8	189.8	*3.5%*	3.6
R112	16.7%	646.4	629.9	*2.5%*	20.8	203.9	190.2	*6.7%*	5.4
vrpnc8	27.0%	581.0	548.1	*5.7%*	50.7	222.0	206.5	*7.0%*	12.9
Average	*19.9%*	*663.3*	*643.2*	*3.2%*	*28.9*	*207.6*	*195.5*	*5.7%*	*7.3*

Based on the computational results presented in Table 7.10, LTA seems to provide savings over all reported instances, with routing cost reductions of up to 5.7% with respect to the complete solution. The savings reported for the transferred pairs present similar behavior to the one reported in Section 7.5.2 (on the average), especially for the NoTW case (vrpcn8), considering that in the current scenario, each route comprises of 4 delivery orders on the average. The results also indicate that as TW width increases, LTA is able to identify more candidate pairs for transfer ($\%\hat{P}$ value). This also leads to improved results. The results are also illustrated, perhaps more clearly, in Figure 7.14. From this Figure it is clear that the performance of LTA improves significantly when the TW width increases from medium (R109) to large (R112) or none (vrpnc8).

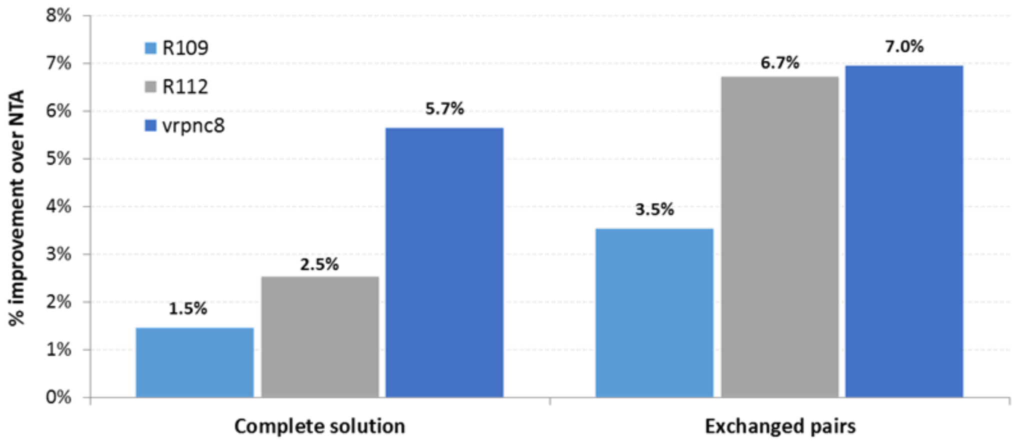


Figure 7.14. Average performance of LTA on the full re-optimization problem

7.5.4 Performance of re-optimization strategies in DVRPMB-LT

In this Section, we evaluate the performance of LTA compared to NTA under different re-optimization policies. To do so, we employed the 100-customer instance without TW of Section 7.5.3 (*vrpnc8*). Based on this instance, we generated 5 different test problems (different selection of delivery orders) by randomly selecting 50% of the customers to be delivery orders. The remaining 50% customers form the set of DO. Each DO was assigned with a time of arrival during the window $[0{,}0.75 * T_{max}]$ according to a continuous uniform distribution. The initial solutions (routes) were obtained with the same process described in Section 5.3 (note that initial solutions involved between 5 and 6 vehicle routes).

For the experimental analysis, we employed the SRR and NRR policies, described in Chapter 5 (see Section 5.2). Recall that in SRR the re-optimization problem is solved upon the arrival of each DO, while in NRR re-optimization is performed after the arrival of a predefined number N of DO. For the latter, we used $N = 0.1\widehat{N}, 0.2\widehat{N}, 0.33\widehat{N}$ (where $\widehat{N}$ is the total number of DO) hereafter designated as NRR-1, NRR-2 and NRR-3. Each policy was tested under the partial-release tactic, i.e. only the DO scheduled for implementation prior to the next re-optimization cycle are released for implementation; the others are re-considered in the following re-optimization cycle. Each re-optimization problem was solved following the framework presented in Section 7.4.1 and only the $\mathrm{LTA_d}$ algorithm was employed.

In order to assess the performance of LTA and NTA under the different re-optimization policies, we employed the so-called *value of information (VOI)* metric (Mitrovic-Minic *et al.*, 2004), described in Chapter 5 (see Section 5.4.1.1) which measures the percentage deviation of the dynamic problem's solution compared to the solution of its static counterpart (i.e. when all DO are known prior to vehicles are dispatched from the depot).

Table 7.11 summarizes the results obtained from solving the dynamic test problems by each re-optimization policy using the NTA and LTA re-optimization algorithms. Specifically, the Table presents the average VOI for all five test problems per policy and re-optimization algorithm (VOI_* designates the average VOI for the * algorithm). The percentage improvement in the last column is the relative percentage improvement of the VOI resulting from the load transfer approach; i.e. $-\left(\frac{VOI_{LTA}-VOI_{NTA}}{VOI_{NTA}}\right)x100.$

Table 7.11. Performance of LTA and NTA per re-optimization policy

Re-optimization Policy	$\mathbf{VOI_{NTA}}$	$\mathbf{VOI_{LTA}}$	%Improvement
SRR	29.3%	26.9%	8.3%
NRR-1	30.2%	27.0%	10.4%
NRR-2	37.0%	31.8%	14.0%
NRR-3	50.0%	40.2%	19.5%

The table shows that LTA improves the results provided by NTA under all re-optimization policies. It is interesting to notice that the percentage improvement increases when the number of elapsed DO per re-optimization cycle increases (less re-optimization cycles). This can be attributed to the fact that infrequent re-optimization causes larger portion of the routes to be completed and allows fewer options available for incorporating newly arrived DO in the current vehicles *en route*. Thus, new vehicles stationed at depot are dispatched in order to cover the demand, causing significant overlaps, which benefit from load transfer operations (LTA). On the other hand, SRR re-considers all DO not yet served providing more possibilities for DO combinations and, thus, better allocation to the available fleet. Thus, load transfer operations may offer limited savings. It should be noted that the percentage improvement (of Table 7.11) in terms of distance travelled ranges from 1.9% (for SRR policy) to 6.5% (for NRR-3 policy) on the average.

Chapter 8: CONCLUSIONS AND FUTURE RESEARCH

8.1 Conclusions

The Dynamic Vehicle Routing Problem with Mixed Backhauls (DVRPMB) seeks to assign in the most efficient way dynamic pick-up requests that arrive in real-time while a predefined distribution plan is being executed. We addressed the DVRPMB through iterative re-optimization. In addition to defining the re-optimization model and appropriate solution methods, we drilled-down to significant aspects concerning the re-optimization process, we addressed the case of limited fleet (in which not all dynamic orders may be served), and the case in which delivery orders are allowed to be transferred to other vehicles.

<u>Re-optimization strategies for DVRPMB</u>

One of the critical elements for tackling DVRPMB concerns the process of updating the *a priori* plan. In this research we considered two fundamental issues: a) the *re-optimization problem* (*how* to re-optimize), and b) *the re-optimization process*; i.e. *when* to re-plan (re-optimization frequency) and *what* part of the new plan to communicate to the drivers (implementation tactic).

Regarding "*how to re-optimize*", we proposed a Branch-and-Price (B&P) approach, which exploits the characteristics of the dynamic problem in hand to solve multiple sub-problems of limited size in order to identify columns that can further enhance the value of the objective function. This allows the algorithm to address re-optimization problems of practical size. Additionally, we appropriately enhanced the dominance criteria used in solving the sub-problems to discard non-promising paths without compromising optimality. For challenging cases (e.g. without time-windows), we proposed a novel insertion heuristic operating within a column generation framework. The latter provides efficient solutions with a limited deviation from the optimum (2.2% on the average).

Regarding the "*when to re-optimize*", we presented and analyzed typical re-optimization policies, i.e.: i) re-optimization upon arrival of each DO, ii) re-optimization after a certain number of DO have been received. In addition, we investigated the effect of two implementation tactics regarding the *"what part"* to release for implementation: i) immediate release of all DO for implementation (FR) and, ii) release of only those DO that are scheduled for implementation prior to the next re-optimization cycle (PR)[11]. We provided theoretical insights regarding the expected behavior of those tactics.

We illustrated through extensive experimentation that re-optimization upon the arrival of each DO under the PR tactic provides superior results on the average. However, this policy seems to be the least favorite when the FR tactic is employed. Furthermore, our experimentation under various operating scenarios has indicated the following: i) when the business case allows it, one should always re-optimize under the PR tactic in as short re-optimization intervals as possible. ii) When the FR tactic is required due to the characteristics of the practical environment, one should prefer re-optimization over short to medium intervals for cases of tight to medium TW, and over medium to large intervals for wider TW cases. iii) In environments with strong dynamism, medium interval policies (regardless of tactic) seem to provide the safest option.

The DVRPMB with limited resources (*m*-DVRPMB)

We also investigated the above problem for the case of limited fleet. In this case, we tested three objective functions. In the first alternative ($\check{z}_1$), the primary objective is to maximize the number of served DO; among the solutions with equal number of served DO, the one with the minimum routing cost is the preferred one. We also introduced a second objective function (objective $\check{z}_2$) that accounts for vehicle productivity during each re-optimization cycle. The objective function attempts to maximize the number of orders served (static and dynamic) within the upcoming re-optimization cycle, among the solutions with the same number of served DO. The third objective function (objective $\check{z}_3$) assigns a profit to each order to be served at any future period, but this profit decreases linearly depending on the period (re-optimization cycle) the order is served. For the latter two objectives, the re-optimization time instances have to be predetermined (known *a priori*). To address the *m*-DVRPMB, we proposed the required modifications in both the DVRPMB model and the solution approach (column generation).

[11] The implementation of this tactic depends on the technology used. Typically the driver receives only the DO to be served next.

We investigated initially through experimentation the effectiveness of the re-optimization heuristic under limited resources and objective $\check{z}_1$. We show that the proposed B&P heuristic offers efficient solutions also in this case with average deviation from the optimum in the order of 1.8%. The experimentation also illustrated that the performance of the re-optimization strategies (policies and tactics) is not affected significantly by limiting the available resources.

We also assessed the performance of the three proposed objectives in *m*-DVRPMB with re-optimization occurring at predefined time instances. The results illustrated that objectives $\check{z}_2$ and $\check{z}_3$ (which consider vehicle productivity) can yield higher customer service compared to objective $\check{z}_1$, offering up to about 15% more DO served under certain conditions. In particular, the experiments indicated that objectives $\check{z}_2$ and $\check{z}_3$ are more appropriate for cases with increasing time-window widths (e.g. with average TW width greater than 40% of the available working horizon), or cases for which a majority (more than 50-60%) of DO may be served by the available fleet. In cases with narrow TW or limited fleet availability, accounting for vehicle productivity does not seem to help appreciably. Furthermore, with respect to re-optimization strategies (policy and tactic), the results illustrate that objectives $\check{z}_2$ and $\check{z}_3$ perform significantly better than objective $\check{z}_1$ under the FR tactic, and objective $\check{z}_3$ seems to be more efficient for the preferred short re-optimization intervals (i.e. 5-15% of the available working horizon).

The application of the proposed method for the *m*-DVRPMB in a next-day courier service provider compared the performance of the proposed B&P heuristic with a) the performance of the process followed by the dispatchers, and b) that of an insertion-based heuristic proposed by Ninikas *et al.* (2014). The results indicate that our B&P algorithm significantly outperforms both the current planning practices of the courier operator and the heuristic used for comparison.

The DVRPMB with Load Transfers (DVRPMB-LT)

We investigated a challenging variant of DVRPMB that allows transfer of orders between vehicles during plan implementation (real-time). In particular, load transfers are considered in the re-optimization problem, which is solved repeatedly in order to incorporate newly received orders in the plan. Allowing for load transfers adds significant complexity to the problem and needs to be dealt in a fundamentally different way compared to the conventional DVRPMB.

For the underlying re-optimization problem, we developed an appropriate model using an arc-based formulation. To solve re-optimization problems related to practice, we restricted each vehicle to participate to up to one transfer operation, and proposed an efficient heuristic framework. The latter considers all possible vehicle pairs that may benefit from a load transfer operation, and solves the related pair-wise problems with an appropriate heuristic procedure. The latter provides solutions of high quality with a limited deviation from the optimal ones (less than 2% on the average).

Considering the re-optimization problem for a pair of vehicles, our experimental results have indicated that load transfer operations may offer average savings of up to 22% when transfer may take place at the location of any not yet served customer, and up to 14% when a fixed (predefined) transfer location is considered. These savings tend to increase when the number of delivery orders increases or when customers are clustered. For the case of multiple vehicles, the re-optimization savings with load transfers reached 5.7% w.r.t. the no transfer case. These savings tend to increase under wider time-windows.

Considering the full dynamic case, load transfer operations result in significant savings, especially under less frequent re-optimization, in which the possibilities of load transfers increase. Even if re-optimization is performed upon the arrival of each new order (SRR policy), the savings are substantial and in the order of 7%.

8.2 Future research

The DVRPMB studied in this dissertation forms a dynamic variant of the more generic one-to-many-to-one PDPs (1-M-1 PDPs). An interesting research direction is to consider the dynamic counterparts of other relevant problems of this family, including the: i) dynamic version of VRPCB, in which linehaul orders (deliveries) must be served prior to backhaul orders (pick-ups) and ii) the dynamic version of VRP with Simultaneous Pickup and Delivery Demands. The performance of the various re-optimization strategies (policies and tactics) proposed in this dissertation may be assessed within these contexts.

A second interesting extension of the current work is to study probabilistic models that consider historical data in order to forecast dynamic demand. Such a model may be combined with the proposed re-optimization process in order to select the appropriate re-optimization policy (i.e. dynamically adapt the re-optimization frequency according to the expected arrival pattern of DO) and tactic (selectively release DO for implementation, regardless of the time they are scheduled to be served).

A related consideration in this dynamic setting concerns the prioritization of orders at each re-optimization cycle. Specifically, during a certain re-optimization cycle it may be beneficial to favor the service of certain customer orders (e.g. urgent ones) in the expense of others, under the assumption that the excluded (e.g. not urgent) ones can fit in the plan during a subsequent re-optimization cycle. Thus, one should examine whether it is beneficial to prioritize service of certain orders, and if so, under which conditions this is favorable to the problem's objective. This consideration can be also be beneficial when forecasting information is available and may be used to prioritize certain orders appropriately.

In the current research, it has been assumed that vehicles are dispatched according to the problem needs, i.e. in order to respond to the demand (customer orders) as it is known at the time of re-optimization. An interesting direction, which is also relevant in practice, is to study different *dispatching policies*, i.e. dispatch more vehicles than necessary (or all vehicles available at depot) at the start of operations or during execution (re-optimization), in anticipation of additional work to come. The performance of these dispatching policies may be studied with respect to various factors of the environment (e.g. degree of dynamism, time-window profiles, etc.), or under the various re-optimization strategies. Those dispatching policies can be investigated in combination with waiting strategies (Mitrovic-Minic and Laporte; 2004, Ichoua *et al.*; 2006) and with stochastic methods that exploit knowledge about future demands (Ichoua *et al.*; 2006).

Regarding the solution of the re-optimization problem, we have focused in this dissertation on column-generation-based algorithms, since we could exploit the structure of the problem in hand and offer near-optimal solutions in reasonable times. However, advanced heuristics or metaheuristics may be investigated targeting faster computational times that can be scalable to the problem size.

Proposals for future research in the interesting case of load transfer operations include the following:

- Following the formulation we developed in Chapter 7, a set-partitioning problem can be also formulated (route-based) in order to develop more efficient algorithms of the column generation type (B&P). As proposed by Cortes *et al.* (2010), the set-partitioning problem could be formulated by introducing additional columns when the transfer location is introduced. Feasibility of the route sequence in order to build a complete trip will be ensured by appropriate binary variables. This approach is expected to permit column generation algorithms to deal with cases of large size.
- In this dissertation, for the solution of the re-optimization problem we have assumed that each vehicle is allowed to participate in only one transfer operation throughout its remaining

(not executed) route. Relaxing this assumption and allowing multiple transfers per vehicle (*one-to-many* policy) could potentially lead to lower costs. Development of more efficient algorithms to handle this context, is encouraged as part of future research.

- The investigation of load transfer operations along with diversion strategies (Ichoua *et al.*; 2000, see also Section 2.3.3.3) may also be an interesting topic of future research. Diverting a vehicle away from its current destination to meet another vehicle operating in the vicinity, may offer higher savings in the total distance traveled.
- More practical aspects for future research on load transfer operations may involve: a) examining the behavior of the algorithm and the overall benefit of load transfers for heterogeneous fleets (those with vehicles of different capacities) and b) applying penalties reflecting the duration of the load-transfer operations, as well as the related interruption, in order to ensure that load-transfers will not be performed in the expense of valuable resources (e.g. time).

REFERENCES

Aas, B., Gribkovskaia, I., Halskau, Ø., Shlopak, A. (2007). Routing of supply vessels to petroleum installations, International Journal of Physical Distribution and Logistics Management 37(2), pp. 164–179.

Angelelli, E., Bianchessi, N., Mansini, R., Speranza, M.G. (2009). Short Term Strategies for a Dynamic Multi-Period Routing Problem, Transportation Research Part C 17(2), pp. 106-119

Anily, S., Hassin, R. (1992). The swapping problem. Networks, 22(4), 419-433.

Anily, S., Gendreau, M., Laporte, G. (2011). The preemptive swapping problem on a tree, Networks 58(2), pp. 83-94.

Attanasio, A., Bregman, J., Ghiani, G., Manni, E. (2007). Real-time fleet management at Ecourier Ltd, In: Zeimpekis, V., Tarantilis, C.D., Giaglis, G.M., Minis, I. (eds.), Dynamic Fleet Management, Operations Research/Computer Science Interfaces, vol. 38, Springer, US, 2007, pp. 219–238.

Attanasio, A., Cordeau, J.F., Ghiani, G., Laporte, G. (2004). Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem, Parallel Computing 30(3), pp. 377–387.

Athanasopoulos, T. (2011). The multi-period vehicle routing problem and its applications, Ph.D. Thesis, Department of Financial & Management Engineering, University of the Aegean, Chios, Greece.

Athanasopoulos, T., Minis, I. (2011). Multi-period routing in Hybrid Courier Operations, In: Minis, I., Zeimpekis, V., Dounias, G., Ampazis, N. (Eds.), Supply Chain Optimization, Design & Management: Advances & Intelligent Methods. IGI Global Publications, pp. 232-251.

Athanasopoulos, T., Minis, I. (2013). Efficient techniques for the multi-period vehicle routing problem with time windows within a branch and price framework, Annals of Operations Research 206(1), pp. 1-22.

Azi, N., Gendreau, M., Potvin, J.Y. (2012). A dynamic vehicle routing problem with multiple delivery routes, Annals of Operations Research, 199(1), pp. 103-112.

Badeau, P., Gendreau, M., Guertin, F., Potvin, J.-Y., Taillard, E.D. (1997). A parallel tabu search heuristic for the vehicle routing problem with time windows, Transportation Research C 5(2), pp. 109–122.

Baker, E., Schaffer, J. (1986). Computational experience with branch exchange heuristics for vehicle routing problems with time window constraints, American Journal of Mathematical and Management Sciences 6, pp. 261–300.

Bard, J.F., Kontoravdis, G., Yu G. (2002). A branch-and-cut procedure for the vehicle routing problem with time windows, Transportation Science 36(2), pp. 250–269.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H. (1998). Branch-and-price: Column generation for solving huge integer programs, Operations Research 46(3), pp. 316–329.

Beaudry, A., Laporte, G., Melo, T., Nickel, S. (2010), Dynamic transportation of patients in hospitals, OR Spectrum 32(1), pp. 77–107.

Bent, R. W., Van Hentenryck P. (2004). Scenario-based planning for partially dynamic vehicle routing with stochastic customers, Operations Research 52(6), pp. 977–987.

Benyahia, I., Potvin, J.Y. (1998). Decision support for vehicle dispatching using genetic programming, IEEE Transactions on Systems Man and Cybernetics Part A – Systems and Humans 28(3), pp. 306–314.

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., Laporte, G. (2007). Static pick-up and delivery problems: A classification scheme and survey, TOP 15(1), pp. 1–31.

Berbeglia, G., Cordeau, J.-F., Laporte, G. (2010). Dynamic pick-up and delivery problems, European Journal of Operational Research 202(1), pp. 8–15.

Berger, J., Barkaoui, M. Braysy, O. (2003). A route-directed hybrid genetic approach for the vehicle routing problem with time windows, INFOR 41(2), pp. 179–194

Bertsimas, D.J., Simchi-Levi, D. (1996). A new generation of vehicle routing research: Robust algorithms addressing uncertainty, Operations Research 44(2), pp. 286–304.

Bertsimas, D., Van Ryzin, G. (1991). A stochastic and dynamic vehicle routing problem in the Euclidean plane, Operations Research 39(4), pp. 601-615.

Bianchessi, N., Righini, G. (2007). Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery, Computers & Operations Research 34(2), pp. 578–594.

Bieding, T., Görtz, S., Klose, A. (2009). On-line routing per mobile phone: a case on subsequent deliveries of newspapers, In: Beckmann, M., Künzi, H.P., Fandel, G., Trockel, W., Basile, A., Drexl, A., Dawid, H., Inderfurth, K., Kürsten, W., Nunen, J.A., Speranza, M.G., Bertazzi L. (eds.), Innovations in Distribution Logistics, Lecture Notes in Economics and Mathematical Systems, Vol. 619, Springer, Berlin Heidelberg, pp. 29–51.

Borodin, A., El-Yaniv, R. (2005). Online Computation and Competitive Analysis, Cambridge University Press, Cambridge.

Bramel, J., Simchi-Levi, D., (2002). Set-covering-based algorithms for the capacitated VRP, In: Toth, P., Vigo, D. (eds), The Vehicle Routing Problem, SIAM, Philadelphia, pp. 85-108.

Brandão, J. (2006). A new tabu search algorithm for the vehicle routing problem with backhauls, European Journal of Operational Research 173(2), pp. 540–555

Branke, J., Middendorf, M., Noeth, M., Dessouky, M. (2005). Waiting Strategies for Dynamic Vehicle Routing, Transportation Science 29(3), pp. 298–312.

Braysy, O. (2002). Fast local searches for the vehicle routing problem with time windows, INFOR 40(4), pp. 319–330.

Braysy, O. (2003). A reactive variable neighborhood search for the vehicle routing problem with time windows, INFORMS Journal on Computing 15(4), pp. 347–368.

Braysy, O., Hasle, G., Dullaert, W. (2004). A multi-start local search algorithm for the vehicle routing problem with time windows, European Journal of Operational Research 159(3), pp. 586-605.

Braysy, O., Gendreau, M. (2005a). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms, Transportation Science 39(1), pp. 104–118.

Braysy, O., Gendreau M. (2005b). Vehicle routing problem with time windows, Part II: Metaheuristics, Transportation Science 39(1), pp. 119–139.

Campbell, A., Savelsbergh, M. (2005). Decision support for consumer direct grocery initiatives, Transportation Science 39(3), pp. 313–327.

Caramia, M., Italiano, G., Oriolo, G., Pacifici, A., Perugia, A. (2002). Routing a fleet of vehicles for dynamic combined pick-up and deliveries services, in: Proceedings of the Symposium on Operation Research 2001, Duisburg, Germany, pp. 3–5.

Casco, D.O., Golden, B.L., Wasil, E.A. (1988). Vehicle routing with backhauls: models, algorithms, and case studies. In: Golden, B.L., Assad, A.A. (eds). Vehicle routing: methods and studie,. North-Holland, Amsterdam, pp. 127–147

Chabrier, A. (2006). Vehicle routing problem with elementary shortest path based column generation, Computers & Operations Research 33(10), pp. 2972–2990.

Chang, M.S., Chen, S.R, Hsueh, C.F. (2003). Real-time vehicle routing problem with time windows and simultaneous delivery/pick-up demands, Journal of the Eastern Asia Society for Transportation Studies 5, pp. 2273-2286.

Chaovalitwongse, W., Kim, D., Pardalos, P.M. (2003). GRASP with a New Local Search Scheme for Vehicle Routing Problems with Time Windows, Journal of Cominatorial Optimization 7(2), pp. 179-207.

Chen, H.K., Hsueh, C.F., Chang, M.S. (2006). The real-time time-dependent vehicle routing problem, Transportation Research Part E: Logistics and Transportation Review 42(5), pp. 383–408.

Chen, Z.L., Xu, H. (2006). Dynamic column generation for dynamic vehicle routing with time windows, Transportation Science 40(1), pp. 74–88.

Cheung, B.K.S., Choy, K.L., Li, C.L., Shi, W., Tang, J. (2008). Dynamic routing model and solution methods for fleet management with mobile technologies, International Journal of Production Economics 113(2), pp. 694–705.

Chiang, W.C., Russell, R.A. (1996). Simulated annealing metaheuristics for the vehicle routing problem with time windows. Annals of Operations Research 63(1), pp. 3–27.

Chiang, W.C., Russell, R.A. (1997). A reactive tabu search metaheuristic for the vehicle routing problem with time windows, INFORMS Journal on Computing 9(4), pp. 417–430.

Christofides, N., Mingozzi, A., Toth, P. (1979). The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., Sandi C. (eds) Combinatorial Optimization, John Wiley, Chichester, pp. 315-339.

Clarke, G., Wright, J. (1964). Scheduling of vehicles from a central depot to a number of delivery points, Operations Research 12(4), pp. 568-581.

Coltin, B., Veloso, M. (2012). Optimizing for transfers in a multi-vehicle collection and delivery problem, In Proceedings of the International Symposium on Distributed Autonomous Robotic Systems (DARS), Baltimore, Maryland.

Cordeau, J.F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F. (2002). VRP with time windows. In: Toth, P., Vigo, D. (eds) The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications vol 9. SIAM, Philadelphia, PA, pp. 175–193.

Cordeau, J.-F., Laporte, G., Mercier, A. (2004). Improved Tabu Search Algorithm for the Handling of Route Duration Constraints in Vehicle Routing Problems with Time Windows, Journal of Operational Research Society 55(3), pp. 542-546.

Cordeau, J.-F., Laporte, G., Potvin, J.-Y., Savelsbergh, M.W.P., (2007). Transportation on demand. In: Barnhart, C., Laporte, G. (Eds.), Handbooks in Operations Research and Management Science: Transportation, vol. 14. North-Holland, Amsterdam, pp. 429–466.

Cordone, R., Calvo, R.W. (1997). A heuristic for vehicle routing problem with time windows. Internal Report 97.012, Politechnico di Milano, Dipartimento di Electronica e Informazione, Milan, Italy.

Cornuejols, G., Nemhauser, G.L. (1978), "Tight bounds for Christofides traveling salesman heuristic", Mathematical Programming 14(1), pp. 116-121.

Cortes, C. E., Matamala, M., Contardo, C. (2010). The pick-up and delivery problem with transfers: Formulation and a branch-and-cut solution method, European Journal of Operational Research 200(3), pp. 711–724.

Danna, E., Le Pape, C. (2005). Branch-and-Price heuristics: A case study on the vehicle routing problem with time windows. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.). Column Generation, Springer Science and Business Media, Inc.

Dantzig, G., Ramser, J. (1959). The Truck Dispatching Problem. Management Science 6(1), pp. 80-91.

Deif, I., Bodin, L. (1984). Extension of the Clarke and Wright algorithm for solving the vehicle routing problem with backhauling. In: Kidder, A. (ed). Proceedings of the Babson conference on software uses in transportation and logistic management, Babson Park, FL

Derigs, U., Metz, A. (1992). A matching-based approach for solving a delivery/pick-up vehicle routing problem with time constraint, Operations Research Spektrum 14(2), pp. 91–106.

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, Laporte, G. (eds) Fleet Management and Logistics. Kluwer Academic Publisher, Boston, Dordrecht, London, pp. 57–93.

Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M.M., Soumis, F. (2002). Vehicle routing problem with pick-up and delivery. In: Toth, P., Vigo, D. (eds) The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications vol 9. SIAM, Philadelphia, PA, pp. 225–242

Desaulniers, G., Desrosiers, J., Solomon, M.M. (eds) (2005) Column Generation, No. 5. In: GERAD 25th Anniversary, Springer

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, pp. 322-332.

Desrochers, M., Desrosiers, J., Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. Operations Research 40(2), pp. 342–354.

Desrosiers, J., Lübbecke, M. (2005). A primer in column generation. In Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.), Column Generation, pp. 1–32, Springer, New York.

Duhamel C, Potvin J-Y, Rousseau JM (1997) A tabu search heuristic for the vehicle routing problem with backhauls and time windows. Transportation Science 31(1), pp. 49–59.

Dondo, R., Mendez, C.A., Cerda, J. (2009). The supply-chain pick-up and delivery problem with transshipments, Computer Aided Chemical Engineering 26, pp. 1009-1014.

Fabri, A., Recht, P. (2006). On dynamic pick-up and delivery vehicle routing with several time windows and waiting times, Transportation Research Part B: Methodological 40(4), pp. 335–350.

Feillet, D., Dejax, P., Gendreau, M., Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems, Networks 44(3), pp. 216-229.

Feillet, D., Gendreau, M., Rousseau, L.M. (2005). New refinements for the solution of vehicle routing problems with branch and price, Technical Report C7PQMR PO2005-08-X, Center for Reasearch on Transportation, Montreal.

Ferrucci, F., Bock, S., Gendreau, M. (2013). A pro-active real-time control approach for dynamic vehicle routing problems dealing with the delivery of urgent goods, European Journal of Operational Research 225(1), pp. 130–141.

Fiegl, C., Pontow, C. (2009). Online scheduling of pick-up and delivery tasks in hospitals, Journal of Biomedical Informatics 42(4), pp. 624–632

Fisher, M.L. (1994). Optimal solution of vehicle routing problems using minimum k-trees, Operations Research 42(4), pp. 626–642.

Fisher, M.L., Jornsten. K.O., Madsen, O.B.G. (1997). Vehicle routing with time windows – Two optimization algorithms, Operations Research 45(3), pp. 488–492.

Fleischmann, B., Gnutzmann, S., Sandvoss, E. (2004). Dynamic vehicle routing based on online traffic information, Transportation Science 38(4), pp. 420–433.

Gambardella, L.M., Taillard, E.D., Agazzi G. (1999). MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows, in: Corne, D., Dorigo, M., Glover F. (eds.), New Ideas in Optimization, McGraw-Hill, London, pp. 63–76.

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, pp. 3-9.

Gelinas, S., Desrochers, M., Desrosiers, J., Solomon, M. M. (1995). A new branching strategy for time constrained routing problems with application to backhauling. Annals of Operations Research 61(1), pp. 91–100.

Gendreau, M., Laporte, G., Hertz, A., (1997). An approximation algorithm for the Traveling Salesman Problem with Backhauls, Operations Research, 45(4), pp. 639-641.

Gendreau, M., Potvin J. Y. (1998). Dynamic Vehicle Routing and Dispatching. In: Crainic, T.G., Laporte, G. (eds), Fleet Management and Logistics, Kluwer, Boston, pp. 115-126.

Gendreau, M., Guertin F., Potvin J. Y., Taillard É. (1999). Parallel tabu search for real-time vehicle routing and dispatching, Transportation Science 33(4), pp. 381–390.

Gendreau, M., Guertin, F., Potvin, J.-Y., Séguin, R. (2006). Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries, Transportation Research Part C 14, pp. 157–174.

Gendreau, M., Potvin, J.Y., (Eds.). Transportation Science. Number 38. (special issue on real-time fleet management), 2004.

Ghiani G., Guerriero F., Laporte G., Musmanno R. (2003). Real-time vehicle routing: solution concepts, algorithms and parallel computing strategies, European Journal of Operational Research 151(1), pp. 1–11.

Ghiani, G., Manni, E., Quaranta, A., Triki, C., (2009). Anticipatory algorithms for same-day courier dispatching, Transportation Research Part E 45(1), pp. 96-106.

Goel, A. (2008). Fleet Telematics: Real-time management and planning of commercial vehicle operations, Operations Research/Computer Science Interfaces Series, Vol. 40, Springer-Verlag International, US.

Goetschalckx, M., Jacobs-Blecha, C. (1989). The vehicle routing problem with backhauls, European Journal of Operational Research 42(1), pp. 39 –51.

Golden, B., Raghavan, S., Wasil, E. (Eds.). (2008). The Vehicle Routing Problem: Latest Advances and New Challenges. Operations Research/Computer Science Interfaces Series, Springer.

Golden, B.L, Assad, A.A. (Eds.). (1988). Vehicles Routing: Methods and Studies. North-Holland, Amsterdam.

Gørtz, I.L., Nagarajan V., Ravi, R. (2009). Minimum makespan multi-vehicle dial-a-ride, In: Proceedings of the 17th ESA, Lecture Notes in CS, Vol. 5757, Copenhagen, Denmark, pp. 540-552.

Gribkovskaia, I., Halskau, Ø., Laporte. G., Vlcek, M. (2007). General solutions to the single vehicle routing problem with pick-ups and deliveries, European Journal of Operational Research, 180(2), pp. 568–584.

Gribkovskaia, I., Laporte, G. (2008). One-to-many-to-one single vehicle pick-up and delivery problems. In: Golden, B.L., Raghavan, S., Wasil, E.A. (Eds.), The Vehicle Routing Problem. Latest Advances and New Challenges, Springer, New York, pp. 359–377.

Gribkovskaia, I., Laporte, G., Shyshou, A. (2008). The single vehicle routing problem with deliveries and selective pick-ups, Computers and Operations Research 35(9), pp. 2908–2924.

Güner, A.R., Murat, A., Chinnam, R.B. (2012). Dynamic routing under recurrent and non-recurrent congestion using real-time its information, Computers & Operations Research 39(2), pp. 358–373.

Gutiérrez-Jarpa, G., Marianov, V., Obreque, C. (2009). A single vehicle routing problem with fixed distribution and optional collections. IIE Transactions 41, pp. 1067–1079.

Gutiérrez-Jarpa, G., Desaulniers, G., Laporte, G., Marianov, V. (2010). A branch-and-price algorithm for the Vehicle Routing Problem with Deliveries, Selective Pickups and Time Windows, European Journal of Operational Research 206(2), pp. 341-349.

Haghani, A., Jung, S. (2005). A dynamic vehicle routing problem with time-dependent travel times, Computers & Operations Research 32(11), pp. 2959–2986.

Harvey, W.D., Ginsberg, M.L. (1995). Limited Discrepancy Search, In: Proceedings IJ-CAI 1995.

Hasama, T., Kokubugata, H., Kawashima, H. (1998). A heuristic approach based on the string model to solve vehicle routing problem with backhauls, 5th World Congress on Intelligent Transport Systems (ITS), Seoul.

Homberger, J., Gehring, H. (2005). A two-phase hybrid metaheuristic for the vehicle routing problem with time windows, European Journal of Operational Research 162(1), pp.220-238.

Ichoua, S., Gendreau M., Potvin J.Y. (2000). Diversion issues in realtime vehicle dispatching, Transportation Science 34(4), pp. 426–438.

Ichoua, S., Gendreau M., Potvin J.Y. (2003). Vehicle dispatching with time-dependent travel times, European Journal of Operational Research 144(2), pp. 379–396.

Ichoua, S., Gendreau, M., Potvin, J.Y. (2006). Exploiting knowledge about future demands for real-time vehicle dispatching. Transportation Science 40(2), pp. 211–225.

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. (Eds.), Dynamic Fleet

Management, Operations Research/Computer Science Interfaces, Vol. 38, Springer, US, pp. 1–18.

Ioannou, G., Kritikos, M., Prastacos G. (2001). A greedy look-ahead heuristic for the vehicle routing problem with time windows. Journal of the Operational Research Society 52(5), pp. 523–537.

Irnich, S., Desaulniers, G. (2005). Shortest path problems with resource constraints. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.), Column Generation, Springer, New York, pp. 33–65.

Jaillet, P., Wagner, M.R. (2008). Generalized online routing: new competitive ratios, resource augmentation, and asymptotic analyses, Operations Research 56(3), pp. 745–757.

Juliff P. (1990). Program Design (3rd edition). Prentice-Hall, Australia.

Kallehauge, B., Larsen J., Madsen O.B.G. (2006). Lagrangean duality applied to the vehicle routing with time windows. Computers & Operations Research 33(5), pp. 1464-1487.

Kerivin, H.L.M., Lacroix, M., Mahjoub, A.R., Quilliot, A. (2008). The splittable pick-up and delivery problem with reloads, European Journal of Industrial Engineering 2(2), pp. 112–133.

Kilby, P., Prosser, P., Shaw, P. (1998). Dynamic VRPs: A study of scenarios, Technical Report APES-06-1998, School of Computer Science, University of St. Andrews, St. Andrews, Scotland.

Kohl, N., Madsen. O.B.G. (1997). An optimization algorithm for the vehicle routing problem with time windows based on lagrangean relaxation. Operations Research 45(3), pp. 395-406.

Kohl, N., Desrosiers, J., Madsen, O.B.G., Solomon, M.M., Soumis, F. (1999). 2-path cuts for the vehicle routing problem with time windows, Transportation Science 33, pp. 101–116.

Kolen, A., Rinnooy-Kan, A., Trienekens, H. (1987). Vehicle routing with time windows. Operations Research 35(2), pp. 266-274.

Kontoravdis, G., Bard, J. F. (1995). A GRASP for the vehicle routing problem with time windows, ORSA Journal on Computing 7(1), pp. 10–23.

Laporte, G. (1992). The vehicle routing problem: An overview of the exact and approximate algorithms. European Journal of Operation Research 53(3), pp. 345-358.

Laporte, G., Osman, I. H. (1995). Routing problems: A bibliography. Annals of Operations Research 61(1), pp. 227-262.

Larsen, A. (2000). The dynamic vehicle routing problem, Ph.D. Thesis (IMM-PHD-2000-73), Institute of Mathematical Modelling, Technical University of Denmark.

Larsen, A., Madsen O. B. G., Solomon M.M. (2002). Partially dynamic vehicle routing—Models and algorithms, Journal of the Operational Research Society 53(6), pp. 637–646.

Larsen, A., Madsen O. B. G., Solomon M.M. (2004). The *a priori* dynamic travelling salesman problem with time windows, Transportation Science 38(4), pp. 459–572.

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. (eds), Dynamic Fleet Management: Concepts, Systems, Algorithms & Case Studies, Operations Research/Computer Science Interfaces Series, Springer, pp. 19-40.

Larsen, A., Madsen, O. B. G., Solomon, M. M. (2008). Recent developments in dynamic vehicle routing systems, in: The Vehicle Routing Problem: Latest Advances and New Challenges, Operations Research/Computer Science Interfaces, Vol. 43, Springer, US, pp. 199–218.

Larsen, J. (2001). Parallelization of the Vehicle Routing Problem with Time Windows, PhD thesis (IMM-PHD-2001-62), Informatics and Mathematical Modelling, Technical University of Denmark.

Lee, E.K., Mitchell, J.E. (2001). Integer Programming: Branch-and-Bound Methods. In: Encyclopedia of Optimization, Volume II, Kluwer Academic Publishers, pp. 509-519.

Lenstra, J.K., Rinnooy Kan, A.H.G. (1981). Complexity on vehicle routing and scheduling problems, Networks 11(2), pp. 221-227.

Li, J.-Q., Mirchandani, P., Borenstein, D. (2009a). A Lagrangian heuristic for the real-time vehicle rescheduling problem. Transportation Research Part E: Logistics and Transportation Review 45(3), pp. 419-433.

Li, J.-Q., Mirchandani, P., Borenstein, D. (2009b). Real-time vehicle rerouting problems with time windows. European Journal of Operational Research 194(3), pp. 711-727.

Lin, C.K.Y. (2008). A cooperative strategy for a vehicle routing problem with pick-up and delivery time windows, Computers & Industrial Engineering 55(4), pp. 766–782.

Lund, K., Madsen, O.B.G., Rygaard, J.M. (1996). Vehicle routing problems with varying degrees of dynamism, Technical report, IMM Institute of Mathematical Modelling.

Mamassis, K., Minis, I., Dikas, G. (2013). Managing vehicle breakdown incidents during urban distribution of a common product. Journal of the Operational Research Society 64(6), pp. 925-937.

Masson, R., Lehuédé, F., Péton, O. (2011). An adaptive large neighborhood search for the pick-up and delivery problem with transfers, Transportation Science 47(3), pp. 344-355.

Masson, R., Lehuédé, F., Péton, O. (2014). The dial-a-ride problem with transfers, Computers & Operations Research 41, pp. 12-23.

Mester, D., Braysy, O. (2005). Active guided evolution strategies for large scale vehicle routing problem with time windows, Computers & Operations Research 32(6), pp. 1593–1614.

Mester, D., Braysy, O., Dullaert, W. (2007). A multi-parametric evolution strategies algorithm for vehicle routing problems, Expert Systems with Applications 32(2), pp. 508-517.

Miller, C.E., Tucker, A.W., Zemlin, R.A. (1960). Integer programming formulations and travelling salesman problems, Journal of Association for Computing Machinery 7(4), pp. 326–329.

Mingozzi, A., Giorgi, S., Baldacci, R. (1999). An exact method for the vehicle routing problem with backhauls, Transportation Science 33(3), pp. 315–329.

Mitrovic-Minic, S., Krishnamurti, R., Laporte, G. (2004). Double-horizon based heuristics for the dynamic pick-up and delivery problem with time windows, Transportation Research Part B, 38(8), pp. 669–685.

Mitrovic-Minic, S., Laporte, G. (2004). Waiting strategies for the dynamic pick-up and delivery problem with time windows, Transportation Research Part B 38(7), pp. 635–655.

Mitrovic-Minic, S., Laporte, G. (2006). The pick-up and delivery problem with time windows and transshipment, INFOR 44(3), pp. 217–227.

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(4), pp. 327-343.

Mu, Q., Fu, Q., Lysgaard, J., Eglese, R. (2011). Disruption management of the vehicle routing problem with vehicle breakdown. Journal of the Operation Research Society, 2(5), 225-234.

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, pp. 113–119.

Nagy, G., Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries, European Journal of Operational Research 162(1), pp. 126-141.

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

Ninikas G., Athanasopoulos T., Zeimpekis V., Minis I. (2014). Integrated planning in hybrid courier operations, International Journal of Logistics Management 25(3), pp. 611-634.

Ninikas, G., Minis, I. (2014). Reoptimization strategies for a dynamic vehicle routing problem with mixed backhauls, Networks 64 (3), pp. 214 – 231.

Novoa, C., Storer, R. (2009). An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. European Journal of Operational Research 196(2), pp. 509–515.

Osman, I. H., Wassan, N. A. (2002). A reactive tabu search meta-heuristic for the vehicle routing problem with back-hauls, Journal of Scheduling 5(4), pp. 263-285

Parragh, S., Doerner, K., Hartl, R. (2008a). A survey on pick-up and delivery problems. Part I: Transportation between customers and depot, Journal für Betriebswirtschaft, 58(1),pp. 21-51.

Parragh, S., Doerner, K., Hartl, R. (2008b). A survey on pick-up and delivery problems. Part II: Transportation between pick-up and delivery locations, Journal für Betriebswirtschaft 58(1), pp. 81-117.

Petersen, H.L., Ropke, S. (2011). The pick-up and delivery problem with cross-docking opportunity, In: Proceedings of the Second international conference on Computational logistics, ICCL'11, Springer-Verlag, Berlin, Heidelberg, pp. 101-113.

Pillac, V., Gendreau M., Guéret, C., Medaglia, A. L. (2013). A review of dynamic vehicle routing problems, European Journal of Operational Research 225(1), pp. 1–11.

Pisinger, D., Ropke, S. (2007). A general heuristic for vehicle routing problems, Computers & Operations Research 34(8), pp. 2403-2435.

Potvin, J.-Y., Rousseau, J.-M. (1993). A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. European Journal of Operational Research, 66(3), pp. 331–340.

Potvin, J.-Y., Rousseau, J.-M. (1995). An exchange heuristic for routing problems with time windows, Journal of the Operational Research Society 46(2), pp. 1433–1446.

Potvin. J-Y, Duhamel, C., Guertin, F. (1996). A genetic algorithm for vehicle routing with backhauling, Applied Intelligence 6(4), pp. 345–355.

Potvin, J.Y., Xu, Y., Benyahia, I. (2006). Vehicle routing and scheduling with dynamic travel times, Computers & Operations Research 33(4), pp. 1129–1137.

Powell, W. B. (1996). A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers, Transportation Science 30(3), pp. 195–219.

Privé, J., Renaud, J., Boctor, F.F., Laporte, G. (2006). Solving a vehicle routing problem arising in soft-drink distribution, Journal of the Operational Research Society 57(9), pp. 1045–1052.

Psaraftis, H.N. (1980). A dynamic-programming solution to the single vehicle many-to-many immediate request dial-a-ride problem, Transportation Science 14(2), pp. 130–154.

Psaraftis, H.N. (1988). Dynamic vehicle routing problems. In: Golden, B.L., Assad, A.A. (Eds.), Vehicle Routing: Methods and Studies. North-Holland, Amsterdam, pp. 223–248.

Qu, Y., Bard, J.F. (2012). A GRASP with adaptive large neighborhood search for pick-up and delivery problems with transshipment, Computers & Operations Research 39(10), pp. 2439-2456.

Reimann, M., Ulrich, H. (2006). Comparing backhauling strategies in vehicle routing using ant colony optimization. Central European Journal of Operations Research 14(2), pp. 105–123.

Reimann, M., Doerner, K., Hartl, R. F. (2002). Insertion based ants for vehicle routing problems with backhauls and time windows, Lecture Notes in Computer Science 2463, pp. 135–148.

Rizzoli, A., Montemanni, R., Lucibello, E., Gambardella, L. (2007). Ant colony optimization for real-world vehicle routing problems, Swarm Intelligence 1, pp. 135–151.

Ropke, S., Pisinger, D. (2006). A unified heuristic for a large dataset of vehicle routing problems with backhauls. European Journal of Operations Research 171(3), pp. 750–775.

Ropke, S., Cordeau, J.-F., Laporte, G. (2007). Models and Branch-and-Cut Algorithms for Pickup and Delivery Problems with Time Windows, Networks 49(4), pp. 258-272.

Rousseau, L.M., Gendreau, M., Pesant, G. (2002). Using constraint-based operators to solve the vehicle routing problem with time windows, Journal of heuristics, 8(1), pp. 43-58.

Russell, R.A. (1977). An effective heuristic for the m-tour traveling salesman problem with some side conditions. Operations Research 25(3), pp. 517–524.

Russell, R.A. (1995). Hybrid heuristics for the vehicle routing problem with time windows. Transportation Science 29(2), pp. 156–166.

Salhi, S., Nagy, G. (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling, Journal of Operational Research Society 50, pp. 1034–1042.

Savelsbergh, M.W.P. (1985). Local search in routing problems with time windows. Annals of Operations Research 4(1), pp. 285–305.

Savelsbergh, M.W.P. (1990). An efficient implementation of local search algorithms for constrained routing problems. European Journal of Operational Research, 47(1), pp. 75–85.

Savelsbergh, M.W.P. (1992). The vehicle routing problem with time windows: Minimizing route duration. ORSA Journal on Computing, 4, pp. 146–154.

Savelsbergh, M.W.P., Sol, M. (1998). DRIVE: Dynamic routing of independent vehicles, Operations Research 46(4), pp. 474–490.

Secomandi, N., Margot, F. (2009). Reoptimization approaches for the vehicle-routing problem with stochastic demands, Operations Research 57(1), pp. 214–230.

Shang, J.S., Cuff, C.K. (1996). Multicriteria pick-up and delivery problem with transfer opportunity, Computers & Industrial Engineering 30(4), pp. 631–645.

Shieh, H. M., May, M. D. (1998). On-line vehicle routing with time windows, optimization-based heuristics approach for freight demands requested in real-time. Transportation Res. Record 1617, pp. 171–178.

Sleator, D., Tarjan, R. (1985). Amortized efficiency of list update and paging rules, Communications of the ACM 28(2), pp. 202–208.

Solomon, M.M. (1986). On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints. Networks, 16(2), pp. 161–174.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints, Operational Research 35(2), pp. 254–265.

Solomon, M.M., Baker, E., Schaffer, J. (1988). Vehicle routing and scheduling problems with time window constraints: Efficient implementations of solution improvement procedures. In: Golden, B.L., Assad, A.A. (Eds.). Vehicle routing: Methods and studies, North-Holland, Amsterdam, 85–106.

Steward, W.R., Golden, B.L., (1983). Stochastic vehicle routing: A comprehensive approach, European Journal of Operations Research 14(4), pp. 371-385.

Süral, H., Bookbinder, J.H. (2003). The single-vehicle routing problem with unrestricted backhauls, Networks 41(3), pp. 127–136.

Taillard, E., Badeau, P., Gendreau, M., Guertin, F., Potvin, J. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows, Transportation Science 31(2), pp. 170–186.

Taillard, E.D., Gambardella, L.M., Gendreau, M., Potvin, J.Y. (2001). Adaptive memory programming: a unified view of metaheuristics, European Journal of Operational Research 135(1), pp. 1–16.

Tan, K.C., Lee, L.H., Ou, K. (2001). Hybrid genetic algorithms in solving vehicle routing problems with time window constraints, Asia-Pacific Journal of Operational Research 18, 170–186.

Tarantilis, C. D., Anagnostopoulou, A., Repoussis, P. (2013). Adaptive Path Relinking for Vehicle Routing and Scheduling Problems with Product Returns, Transportation Science 47 (3), pp. 356–379.

Tatarakis, A., Minis, I. (2009). Stohastic single vehicle routing with a predefined customer sequence and multi depot returns, European Journal of Operation Research 197(2), pp. 557-571.

Thangiah, S., Potvin, J. Y., Sun, T. (1996). Heuristic approaches to vehicle routing with backhauls and time windows. Computers & Operations Research 23(11), pp. 1043–1057.

Thangiah, S.R., Fergany, A., Awam, S. (2007). Real-time split delivery pick-up and delivery time window problems with transfers, Central European Journal of Operations Research 15(4), pp. 329-349.

Toth, P., Vigo, D. (1997). An exact algorithm for the vehicle routing problem with backhauls. Transportation Science 31(4), pp. 372–385.

Toth, P., Vigo, D. (1999). A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls, European Journal of Operational Research 113(3), pp. 528–543.

Toth, P., Vigo, D. (2002). The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA.

Van Hemert, J.I., Poutré, J.L. (2004). Dynamic routing problems with fruitful regions: models and evolutionary computation, In: Yao, X., Burke, E., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J., Tino, P., Kabán, A., Schwefel, H.P. (Eds.), Parallel Problem Solving from Nature, Lecture Notes in Computer Science, vol. 3242, Springer, Berlin/Heidelberg, pp. 692–701.

Wade, A.C., Salhi, S. (2002). An investigation into a new class of vehicle routing problems with backhauls, Omega 30, pp. 479–487.

Wang, X., Cao, H. (2008). A dynamic vehicle routing problem with backhaul and time window, in Proceedings of the IEEE International Conference on Service Operations and Logistics, and Informatics, IEEE/SOLI 2008, pp. 1256-1261.

Wang, F., Lim, A., Xu, Z. (2006). The one-commodity pick-up and delivery traveling salesman problem on a path or a tree, Networks 48(1), pp. 24–35.

Yang, J., Jaillet, P., Mahmassani, H. (2004). Real-time multi-vehicle truckload pick-up and delivery problems, Transportation Science 38(2), pp. 135–148.

Yano, C., Chan, T., Richter, L., Cutler, T., Murty, K., Mcgettigan, D. (1987). Vehicle routing at Quality Stores, Interfaces 17(2), pp. 52–63.

Zachariadis, E. E., Kiranoudis, C. T. (2012). An effective local search approach for the Vehicle Routing Problem with Backhauls, Expert Systems with Applications 39(3), pp. 3174–3184

Zeimpekis, V., Tarantilis, C. D., Giaglis, G. M., Minis, I. (2007). Dynamic Fleet Management: Concepts, Systems, Algorithms & Case Studies, Operations Research/Computer Science Interfaces Series, Vol. 38, Springer-Verlag International, US.

Zhong, Y., Cole, M. (2005). A vehicle routing problem with backhauls and time windows: A guided local search solution. Transportation Research E 41(2), pp.131–144.

APPENDICES

Appendix A. Detailed results of the experiments presented in Chapter 5

We present here the detailed experimental results, which were summarized in Chapter 5. Specifically, Table A.1 provides additional performance indicators of HEUR and OPT algorithms per investigated dataset, following the results displayed in Table 5.3 of Chapter 5 (see Section 5.4.2); TD refers to total distance traveled and NR refers to number of routes in the final solution. All values are averages w.r.t. all problems and instances.

Tables A.2-A.7 present the detailed results of the re-optimization strategies per instance as illustrated in Section 5.4.3 of Chapter 5. In particular, Tables A.2-A.3, provide the detailed results in terms of VOI, while Tables A.4-A.7 present detailed performance indicators (distance travelled and number of routes) per re-optimization strategy and instance. Note that for those Tables, the values presented are averages w.r.t. all problems (replicates).

Table A.1. Additional performance indicators of HEUR and OPT algorithms

Dataset	Nodes	$dod = 25\%$				$dod = 50\%$			
		HEUR		OPT		HEUR		OPT	
		TD	NR	TD	NR	TD	NR	TD	NR
R1	100	1404.0	15.6	1375.9	15.0	1443.8	16.1	1417.9	15.7
C1	100	958.9	11.1	933.8	10.6	941.1	10.8	917.8	10.5
RC1	100	1573.4	15.1	1534.3	14.6	1597.2	14.7	1565.8	14.7
MR2	50	802.4	4.1	785.9	3.5	816.0	5.0	798.8	4.1
MC2	50	483.4	3.8	476.9	3.6	517.3	3.9	507.6	3.7
MRC2	50	879.3	5.0	855.9	4.3	926.4	5.4	906.0	4.3
Average		***1016.9***	***9.1***	***993.8***	***8.6***	***1040.3***	***9.3***	***1019.0***	***8.8***

Table A.2. Detailed performance of re-optimization strategies for R1, C1 and RC1 instances and for 25% and 75% dod

Instance	$dod = 25\%$								$dod = 75\%$							
	FR				PR				FR				PR			
	SRR	NRR-1	NRR-2	NRR-3	SRR	NRR-1	NRR-2	NRR-3	SRR	NRR-1	NRR-2	NRR-3	SRR	NRR-1	NRR-2	NRR-3
R101	4.8%	4.6%	5.7%	8.9%	3.1%	4.0%	5.5%	8.9%	38.3%	36.2%	49.7%	87.7%	27.2%	32.4%	49.5%	87.7%
R102	14.3%	13.2%	12.4%	19.6%	12.1%	13.1%	12.3%	19.6%	47.2%	42.7%	43.9%	94.1%	45.4%	41.2%	43.4%	94.0%
R103	16.9%	19.0%	17.4%	25.5%	15.0%	18.2%	17.4%	25.5%	42.9%	37.5%	46.8%	81.6%	24.5%	27.7%	36.1%	77.7%
R104	28.8%	31.7%	31.9%	41.1%	25.9%	30.3%	31.9%	41.1%	69.0%	40.9%	45.5%	39.1%	15.4%	14.5%	29.3%	32.6%
R105	8.8%	8.3%	7.1%	13.1%	3.9%	5.8%	6.9%	13.0%	22.6%	17.2%	27.9%	53.3%	5.9%	7.0%	20.2%	54.0%
R106	10.9%	11.1%	13.7%	22.2%	9.0%	10.8%	13.0%	22.2%	64.1%	66.6%	79.9%	139.8%	47.8%	62.3%	79.2%	139.8%
R107	13.4%	15.0%	16.1%	28.7%	10.8%	14.6%	15.5%	28.7%	43.7%	41.1%	47.3%	94.9%	31.0%	28.4%	38.0%	86.7%
R108	31.0%	23.5%	25.9%	46.5%	27.2%	20.4%	25.9%	46.5%	66.2%	45.9%	48.5%	42.1%	16.0%	18.4%	29.3%	35.4%
R109	7.0%	5.7%	5.4%	6.1%	2.4%	2.7%	4.0%	5.6%	41.8%	21.8%	15.2%	27.3%	6.0%	8.3%	11.7%	26.2%
R110	19.0%	18.1%	20.9%	21.4%	13.7%	15.2%	19.2%	21.2%	64.7%	45.9%	54.9%	79.3%	29.0%	28.6%	49.6%	75.7%
R111	17.4%	18.2%	19.7%	25.4%	12.2%	14.7%	17.9%	24.1%	50.4%	45.0%	50.9%	85.9%	24.8%	22.5%	35.7%	80.5%
R112	17.6%	12.7%	11.6%	9.6%	3.8%	4.6%	4.6%	6.3%	95.2%	56.5%	38.0%	25.2%	19.3%	16.9%	15.1%	15.8%
R100	30.6%	28.4%	26.1%	23.6%	15.3%	20.1%	19.9%	19.4%	109.3%	79.1%	87.0%	101.0%	40.5%	44.5%	59.6%	80.9%
C101	6.1%	10.8%	17.7%	44.6%	5.7%	10.7%	17.7%	44.6%	1.4%	46.4%	46.4%	59.7%	1.4%	46.3%	46.3%	51.7%
C102	5.6%	9.4%	15.0%	33.0%	4.5%	8.4%	15.1%	33.0%	71.9%	24.8%	70.6%	86.1%	63.5%	24.4%	69.5%	76.6%
C103	14.3%	16.4%	20.4%	38.1%	12.1%	14.7%	20.8%	38.1%	58.3%	56.8%	118.3%	146.7%	52.3%	21.7%	104.1%	125.6%
C104	33.9%	20.1%	39.9%	54.6%	33.9%	20.1%	39.9%	54.6%	69.4%	63.2%	113.0%	130.2%	55.3%	43.0%	88.0%	98.8%
C105	3.3%	6.9%	17.5%	45.3%	1.6%	6.3%	16.4%	45.3%	44.0%	37.0%	145.9%	154.2%	3.6%	24.5%	136.0%	139.5%
C106	2.5%	9.6%	14.5%	38.9%	2.4%	9.5%	14.3%	38.9%	75.7%	82.9%	149.3%	169.4%	16.6%	56.9%	148.8%	177.4%
C107	21.1%	27.2%	31.4%	58.9%	19.5%	26.7%	30.8%	58.9%	51.3%	49.7%	68.2%	78.8%	17.2%	30.1%	55.7%	66.4%
C108	5.6%	10.3%	9.3%	22.5%	1.7%	6.9%	6.9%	21.7%	65.8%	44.1%	66.7%	74.7%	17.0%	31.5%	62.3%	72.8%
C109	13.9%	16.0%	17.0%	24.5%	8.8%	11.9%	13.4%	23.8%	51.7%	43.3%	61.5%	79.1%	21.1%	26.2%	57.8%	69.3%
C100	21.7%	22.4%	15.7%	23.4%	15.4%	13.2%	14.5%	18.6%	67.4%	65.6%	71.6%	83.2%	55.6%	61.2%	66.5%	74.7%
RC101	7.1%	7.2%	7.9%	10.9%	4.3%	4.2%	6.6%	10.0%	39.1%	31.5%	36.5%	38.1%	22.1%	21.7%	30.5%	34.0%
RC102	8.4%	7.7%	7.7%	10.8%	4.7%	6.7%	6.4%	10.8%	30.8%	27.9%	26.0%	39.0%	19.5%	19.1%	20.3%	39.0%
RC103	8.1%	7.0%	8.4%	7.4%	4.5%	5.0%	7.2%	6.8%	58.3%	50.1%	48.0%	61.4%	18.2%	23.8%	18.8%	44.6%
RC104	12.5%	11.5%	13.0%	21.6%	8.7%	6.7%	12.4%	21.9%	88.6%	60.0%	52.8%	69.1%	21.7%	24.8%	29.6%	37.8%
RC105	12.8%	16.0%	14.3%	17.5%	10.0%	12.7%	12.9%	16.9%	48.1%	35.1%	30.5%	41.8%	18.2%	20.1%	22.9%	29.9%
RC106	13.0%	13.1%	12.7%	11.2%	8.2%	9.9%	9.6%	9.9%	60.4%	40.5%	37.5%	35.0%	22.4%	24.6%	29.9%	31.2%
RC107	17.4%	15.4%	17.5%	18.2%	10.3%	10.3%	11.0%	12.6%	78.3%	48.4%	49.4%	39.8%	23.3%	21.6%	26.2%	28.3%
RC108	24.2%	17.5%	15.6%	12.0%	10.1%	10.5%	7.4%	9.0%	94.8%	60.3%	56.3%	42.3%	25.3%	20.6%	23.3%	29.0%

Table A.3. Detailed performance of re-optimization strategies for R1, C1, RC1, MR2, MC2 and MRC2 instances and for 50% dod

Instance	$dod = 50\%$								Instance	$dod = 50\%$							
	FR				PR					FR				PR			
	SRR	NRR-1	NRR-2	NRR-3	SRR	NRR-1	NRR-2	NRR-3		SRR	NRR-1	NRR-2	NRR-3	SRR	NRR-1	NRR-2	NRR-3
R101	15.9%	15.4%	17.4%	37.3%	9.4%	12.1%	17.0%	37.3%	**MR201**	13.1%	10.1%	7.5%	8.0%	3.0%	2.4%	3.0%	4.0%
R102	30.1%	33.0%	30.8%	56.7%	24.6%	30.2%	30.6%	56.6%	**MR202**	15.9%	11.5%	8.7%	9.0%	1.5%	1.9%	3.8%	5.0%
R103	44.3%	43.9%	41.3%	70.1%	38.9%	42.8%	40.5%	70.0%	**MR203**	12.4%	17.4%	23.7%	24.0%	7.4%	13.8%	22.9%	25.0%
R104	33.3%	30.7%	34.9%	50.4%	25.6%	23.0%	28.9%	48.6%	**MR204**	33.3%	47.7%	42.2%	45.0%	17.2%	27.1%	37.9%	50.0%
R105	11.4%	8.9%	12.2%	18.5%	5.2%	5.1%	9.3%	18.2%	**MR205**	17.9%	14.9%	9.7%	10.0%	4.9%	2.3%	2.8%	6.0%
R106	29.4%	32.7%	33.7%	69.5%	18.5%	26.2%	32.1%	69.5%	**MR206**	21.5%	21.8%	11.7%	15.0%	5.4%	3.9%	3.7%	5.0%
R107	42.5%	35.9%	40.2%	82.7%	31.4%	35.7%	39.6%	82.4%	**MR207**	21.6%	14.3%	8.7%	10.0%	6.2%	3.3%	6.0%	7.0%
R108	37.7%	29.9%	34.2%	54.3%	24.1%	20.3%	28.6%	50.9%	**MR208**	42.2%	42.1%	40.8%	43.0%	21.3%	22.4%	38.5%	35.0%
R109	17.7%	15.6%	12.9%	14.3%	3.8%	4.3%	5.3%	12.9%	**MR209**	19.9%	10.7%	9.9%	10.0%	7.1%	2.2%	2.7%	7.0%
R110	39.6%	39.0%	43.4%	57.3%	27.7%	25.6%	37.6%	56.0%	**MR210**	19.0%	15.5%	9.9%	10.0%	3.4%	2.0%	4.6%	7.0%
R111	36.5%	31.0%	32.7%	53.8%	21.1%	25.2%	27.4%	52.7%	**MR211**	24.9%	25.8%	17.2%	18.0%	10.9%	11.7%	12.0%	13.0%
R112	54.1%	32.2%	25.3%	20.0%	9.9%	10.8%	11.6%	11.5%									
R100	67.5%	56.8%	53.8%	64.9%	29.5%	31.6%	38.6%	52.2%									
C101	14.4%	23.4%	72.2%	88.2%	11.8%	22.0%	72.0%	92.9%	**MC201**	38.2%	21.8%	17.3%	8.8%	4.9%	5.3%	5.0%	5.8%
C102	30.6%	40.7%	62.1%	72.1%	25.7%	38.1%	62.1%	69.5%	**MC202**	25.5%	10.7%	8.5%	5.0%	6.5%	4.8%	3.8%	3.5%
C103	64.1%	30.0%	70.7%	90.9%	62.5%	29.2%	68.8%	77.9%	**MC203**	60.4%	53.9%	47.2%	66.5%	29.7%	24.9%	43.1%	66.8%
C104	57.0%	46.7%	103.9%	119.7%	64.7%	42.5%	89.0%	106.6%	**MC204**	44.1%	51.6%	80.7%	81.4%	10.7%	15.9%	35.2%	42.3%
C105	16.9%	30.2%	70.0%	84.3%	7.9%	26.2%	65.4%	80.3%	**MC205**	70.5%	37.7%	35.4%	26.6%	2.8%	2.7%	4.3%	4.7%
C106	17.4%	27.8%	58.5%	71.7%	5.8%	18.1%	53.1%	59.0%	**MC206**	50.1%	32.1%	21.8%	18.8%	4.7%	5.7%	9.2%	6.6%
C107	35.0%	37.2%	73.8%	84.9%	19.9%	28.4%	66.8%	82.3%	**MC207**	55.8%	31.5%	22.2%	24.1%	3.4%	4.0%	5.8%	7.5%
C108	50.3%	45.2%	37.4%	43.9%	1.1%	5.6%	32.6%	40.9%	**MC208**	33.8%	43.3%	21.1%	15.0%	5.7%	4.4%	2.9%	1.7%
C109	33.5%	34.1%	14.2%	18.3%	7.1%	3.7%	8.2%	9.9%									
C100	48.5%	48.2%	49.0%	54.8%	37.0%	41.8%	46.5%	54.8%									
RC101	24.0%	18.7%	22.2%	36.1%	15.8%	16.1%	19.8%	36.1%	**MRC201**	19.9%	17.1%	13.7%	14.0%	3.9%	1.2%	1.3%	5.0%
RC102	22.4%	17.7%	28.3%	38.5%	9.9%	11.8%	24.7%	38.5%	**MRC202**	30.5%	20.8%	11.3%	15.0%	3.4%	1.4%	1.4%	5.0%
RC103	23.1%	20.0%	22.0%	41.9%	11.6%	13.0%	20.1%	41.8%	**MRC203**	41.9%	29.7%	22.3%	24.0%	14.1%	11.5%	21.9%	25.0%
RC104	34.9%	36.3%	38.6%	36.1%	17.6%	18.2%	18.8%	27.3%	**MRC204**	34.8%	41.3%	60.8%	62.0%	12.6%	27.6%	53.4%	60.0%
RC105	23.7%	24.8%	24.6%	29.0%	12.2%	15.5%	19.1%	23.1%	**MRC205**	24.7%	16.7%	15.6%	20.0%	5.1%	1.8%	2.9%	6.0%
RC106	28.5%	22.5%	19.8%	23.4%	13.7%	14.1%	14.0%	21.9%	**MRC206**	29.5%	18.4%	10.5%	13.0%	3.1%	1.8%	1.4%	5.0%
RC107	33.4%	27.3%	28.6%	29.8%	15.9%	15.9%	20.2%	30.2%	**MRC207**	29.4%	21.6%	14.2%	17.0%	11.4%	3.6%	4.9%	10.0%
RC108	47.8%	30.9%	24.3%	17.8%	10.6%	12.1%	16.5%	13.1%	**MRC208**	33.1%	24.0%	20.1%	25.0%	8.9%	7.4%	10.9%	12.0%

Table A.4. Additional performance indicators of re-optimization strategies for R1, C1 and RC1 instances and for 25% dod

Instance	FR								PR							
	SRR		NRR-1		NRR-2		NRR-3		SRR		NRR-1		NRR-2		NRR-3	
	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR
R101	1840.1	21.3	1836.6	21.4	1855.9	21.9	1912.1	23.6	1810.3	21.1	1826.1	21.4	1852.4	21.9	1912.1	23.6
R102	1847.4	23.8	1829.7	23.8	1816.7	23.9	1933.1	26.8	1811.9	23.5	1828.0	23.9	1815.1	23.9	1933.1	26.8
R103	1700.8	21.2	1731.3	22.3	1708.0	22.6	1825.9	25.7	1673.1	21.2	1719.7	22.1	1708.0	22.6	1825.9	25.7
R104	1583.0	21.6	1618.6	21.5	1621.1	22.6	1734.1	26.3	1547.3	21.5	1601.4	21.3	1621.1	22.6	1734.1	26.3
R105	1663.8	18.2	1656.1	18.5	1637.8	18.4	1729.5	20.3	1588.8	17.4	1617.9	18.0	1634.7	18.5	1728.0	20.3
R106	1555.3	17.3	1558.1	17.2	1594.5	19.0	1713.7	21.7	1528.6	17.2	1553.9	17.4	1584.7	18.9	1713.7	21.7
R107	1478.4	15.4	1499.3	16.6	1513.6	16.9	1677.9	21.2	1444.5	15.1	1494.0	16.7	1505.8	16.8	1677.9	21.2
R108	1500.1	18.6	1414.2	16.4	1441.7	17.9	1677.6	24.9	1456.6	18.3	1378.7	16.6	1441.7	17.9	1677.6	24.9
R109	1427.2	14.0	1409.9	14.0	1405.9	14.2	1415.2	15.2	1365.9	14.4	1369.9	14.3	1387.2	14.5	1408.6	15.1
R110	1495.1	16.1	1483.8	16.6	1519.0	17.6	1525.3	19.2	1428.5	16.2	1447.4	16.4	1497.6	17.6	1522.8	19.3
R111	1472.0	16.9	1482.0	17.2	1500.8	17.6	1572.3	19.6	1406.8	16.3	1438.1	16.8	1478.2	17.4	1556.0	19.4
R112	1330.7	12.0	1275.2	11.8	1262.8	12.2	1240.1	12.3	1174.5	11.5	1183.6	11.9	1183.6	11.7	1202.8	12.3
R100	1317.7	13.4	1295.5	13.9	1272.3	14.3	1247.1	13.9	1163.3	12.2	1211.8	13.6	1209.8	13.4	1204.7	13.3
C101	973.0	12.3	1016.1	13.0	1079.4	14.5	1326.1	18.5	969.3	12.2	1015.2	13.0	1079.4	14.5	1326.1	18.5
C102	1044.4	12.0	1082.0	13.0	1137.3	14.1	1315.4	17.3	1033.5	12.2	1072.1	12.9	1138.3	14.1	1315.4	17.3
C103	1211.2	12.1	1233.5	12.8	1275.9	13.6	1463.4	16.8	1187.9	12.4	1215.5	12.8	1280.1	13.7	1463.4	16.8
C104	1596.3	23.9	1431.8	20.8	1667.9	27.2	1843.1	31.0	1596.3	23.9	1431.8	20.8	1667.9	27.2	1843.1	31.0
C105	928.7	10.9	961.1	11.5	1056.4	12.8	1306.3	17.2	913.4	10.9	955.7	11.5	1046.5	12.8	1306.3	17.2
C106	940.4	11.1	1005.5	12.5	1050.5	13.3	1274.3	17.6	939.5	11.1	1004.6	12.3	1048.6	13.3	1274.3	17.6
C107	1130.1	10.5	1187.0	11.5	1226.2	12.6	1482.8	16.9	1115.2	10.6	1182.4	11.6	1220.6	12.8	1482.8	16.9
C108	973.3	10.4	1016.7	11.0	1007.5	11.5	1129.1	14.1	937.4	10.8	985.3	11.3	985.3	11.5	1121.7	14.1
C109	1067.8	10.0	1087.5	10.0	1096.8	10.3	1167.2	11.5	1020.0	10.4	1049.0	10.4	1063.1	10.5	1160.6	11.6
C100	1326.4	13.1	1334.0	13.6	1261.0	12.9	1344.9	14.5	1257.7	13.9	1233.7	13.3	1247.9	13.1	1292.6	14.0
RC101	1995.0	20.0	1996.9	20.0	2009.9	20.6	2065.8	21.8	1942.8	20.0	1941.0	20.0	1985.7	20.4	2049.0	21.8
RC102	1817.1	18.4	1805.4	18.8	1805.4	18.8	1857.3	19.6	1755.1	18.2	1788.6	18.8	1783.6	18.6	1857.3	19.8
RC103	1653.6	15.6	1636.7	15.8	1658.2	16.0	1642.9	16.2	1598.5	15.6	1606.1	15.6	1639.8	15.6	1633.7	16.2
RC104	1557.2	14.0	1543.3	14.0	1564.1	14.8	1683.1	17.0	1504.6	14.0	1476.9	13.6	1555.8	14.8	1687.3	17.0
RC105	2018.4	20.0	2075.7	20.6	2045.3	20.4	2102.5	21.6	1968.3	19.8	2016.7	20.4	2020.2	20.2	2091.8	21.6
RC106	1755.1	14.8	1756.7	15.6	1750.5	15.2	1727.2	15.2	1680.6	15.2	1707.0	15.6	1702.3	15.4	1707.0	15.2
RC107	1669.2	14.8	1640.8	15.2	1670.6	15.6	1680.6	16.4	1568.2	15.2	1568.2	14.8	1578.2	15.2	1600.9	15.6
RC108	1741.5	14.6	1647.6	14.2	1620.9	14.6	1570.5	13.8	1543.8	14.0	1549.4	13.8	1506.0	13.8	1528.4	13.4

Table A.5. Additional performance indicators of re-optimization strategies for R1, C1 and RC1 instances and for 50% dod

Instance	FR								PR							
	SRR		NRR-1		NRR-2		NRR-3		SRR		NRR-1		NRR-2		NRR-3	
	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR
R101	2472.7	25.1	2435.2	25.7	2676.5	27.1	3356.0	37.3	2274.3	23.7	2367.2	24.6	2673.0	26.7	3356.0	37.3
R102	2407.8	29.1	2334.2	30.1	2353.8	30.3	3175.0	42.0	2378.4	27.6	2309.7	29.5	2345.7	30.2	3173.4	42.0
R103	2037.4	28.8	1960.4	28.7	2093.0	30.7	2589.1	41.3	1775.0	28.0	1820.7	28.1	1940.4	30.5	2533.5	41.2
R104	1890.1	15.4	1575.9	15.9	1627.3	18.5	1555.7	24.4	1290.7	16.9	1280.6	15.7	1446.1	18.2	1483.0	24.3
R105	1887.2	18.3	1804.1	17.8	1968.8	18.9	2359.8	21.8	1630.1	17.8	1647.1	17.9	1850.3	18.7	2370.5	22.1
R106	2285.1	19.9	2319.9	21.5	2505.1	23.2	3339.3	37.2	2058.1	18.7	2260.1	20.2	2495.4	23.0	3339.3	37.2
R107	1806.7	19.9	1774.0	19.6	1852.0	24.2	2450.5	37.6	1647.0	19.1	1614.4	20.1	1735.1	24.0	2347.4	37.5
R108	1820.3	13.2	1597.9	13.9	1626.4	17.2	1556.3	24.2	1270.5	14.8	1296.7	14.0	1416.1	16.9	1482.9	23.3
R109	1902.5	15.2	1634.1	14.8	1545.6	15.3	1707.9	16.5	1422.1	14.6	1453.0	14.2	1498.6	14.7	1693.2	16.1
R110	2025.2	18.7	1794.1	19.0	1904.7	24.2	2204.8	29.4	1586.2	19.1	1581.3	18.0	1839.6	23.6	2160.5	29.3
R111	1872.1	19.1	1804.9	18.6	1878.4	20.8	2314.0	27.5	1553.5	17.7	1524.8	18.6	1689.2	19.9	2246.8	27.1
R112	2280.8	14.1	1828.6	13.5	1612.4	13.8	1462.9	13.9	1393.9	13.3	1365.9	12.9	1344.9	12.5	1353.0	13.0
R100	2154.3	17.9	1843.5	18.0	1924.8	19.6	2068.9	23.7	1446.2	16.0	1487.3	16.2	1642.8	18.5	1862.0	21.5
C101	936.7	12.3	1352.4	14.2	1352.4	22.9	1475.2	24.1	936.7	12.4	1351.5	14.3	1351.5	23.1	1401.3	24.1
C102	1799.3	13.5	1306.3	15.8	1785.7	20.2	1947.9	25.2	1711.4	13.9	1302.1	15.8	1774.2	20.2	1848.5	25.2
C103	1881.0	16.3	1863.2	15.8	2593.9	20.2	2931.4	22.3	1809.7	15.9	1446.1	14.3	2425.2	19.3	2680.7	22.3
C104	1730.8	11.9	1667.5	13.8	2176.3	29.1	2352.0	28.5	1586.7	18.3	1461.1	15.1	1920.9	27.1	2031.2	26.5
C105	1285.1	11.2	1222.7	13.3	2194.6	19.9	2268.6	23.4	924.6	11.1	1111.1	13.8	2106.2	19.8	2137.4	20.3
C106	1629.7	12.2	1696.5	14.7	2312.4	21.2	2498.8	23.6	1081.5	11.8	1455.3	13.8	2307.7	20.9	2573.0	22.7
C107	1424.9	10.1	1409.8	10.3	1584.0	15.8	1683.9	18.9	1103.7	10.1	1225.2	10.9	1466.3	16.7	1567.1	17.6
C108	1402.9	10.5	1219.3	11.0	1410.5	14.7	1478.2	16.1	990.0	10.0	1112.7	10.4	1373.3	14.7	1462.2	15.3
C109	1523.7	10.0	1439.3	10.1	1622.1	11.7	1798.9	16.6	1216.3	11.4	1267.6	10.7	1584.9	11.2	1700.4	17.4
C100	1867.3	16.1	1847.2	16.7	1914.1	17.7	2043.5	20.8	1735.6	16.7	1798.1	16.7	1857.2	17.9	1948.7	20.3
RC101	2551.9	23.2	2412.5	23.0	2504.2	25.0	2533.6	29.4	2240.0	22.2	2232.7	22.4	2394.2	24.4	2458.4	29.4
RC102	2167.6	19.0	2119.6	19.6	2088.1	21.6	2303.5	25.0	1980.4	18.4	1973.7	18.8	1993.6	21.4	2303.5	25.0
RC103	2373.1	15.8	2250.2	16.0	2218.7	18.2	2419.6	23.6	1772.0	14.8	1855.9	15.4	1781.0	18.6	2167.7	23.6
RC104	2372.0	13.8	2012.3	15.2	1921.7	16.2	2126.7	16.6	1530.6	13.8	1569.6	13.4	1630.0	15.0	1733.1	16.2
RC105	2664.3	22.0	2430.5	23.2	2347.7	23.6	2551.0	24.0	2126.4	20.6	2160.6	20.8	2211.0	22.4	2336.9	23.4
RC106	2493.3	17.2	2184.0	16.4	2137.4	16.8	2098.5	19.4	1902.6	16.8	1936.8	16.4	2019.2	17.2	2039.4	19.0
RC107	2535.3	16.2	2110.2	15.8	2124.4	17.2	1987.9	20.4	1753.3	16.4	1729.1	15.6	1794.5	17.0	1824.3	20.6
RC108	2732.1	15.8	2248.2	15.2	2192.1	16.4	1995.8	15.8	1757.3	14.6	1691.4	14.4	1729.3	15.0	1809.2	14.8

Table A.6. Additional performance indicators of re-optimization strategies for R1, C1 and RC1 instances and for 75% dod

Instance	FR								PR							
	SRR		NRR-1		NRR-2		NRR-3		SRR		NRR-1		NRR-2		NRR-3	
	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR
R101	1974.5	30.7	1966.0	32.3	2000.1	32.8	2339.1	41.0	1863.8	30.7	1909.8	31.9	1993.3	32.5	2339.1	41.0
R102	2016.1	31.6	2061.1	30.2	2027.0	30.6	2428.3	34.9	1930.9	31.2	2017.7	30.0	2023.9	30.6	2426.8	34.9
R103	1921.7	19.2	1916.3	19.7	1881.7	24.2	2265.2	39.3	1849.7	20.1	1901.7	20.5	1871.1	23.4	2263.9	38.6
R104	1467.8	15.3	1439.2	14.0	1485.4	20.0	1656.1	19.6	1383.0	13.5	1354.4	13.2	1419.3	18.4	1636.2	18.8
R105	1659.2	19.2	1622.0	18.4	1671.1	21.2	1765.0	34.3	1566.9	17.1	1565.4	17.4	1627.9	20.7	1760.5	34.9
R106	1792.2	28.3	1837.9	29.8	1851.8	29.0	2347.6	41.5	1641.2	26.7	1747.9	29.4	1829.6	28.9	2347.6	41.5
R107	1750.3	18.5	1669.2	18.9	1722.0	22.3	2244.0	38.5	1613.9	19.6	1666.7	18.4	1714.7	21.0	2240.3	36.2
R108	1474.9	14.5	1391.4	13.7	1437.4	19.1	1652.7	19.5	1329.2	12.6	1288.5	13.1	1377.4	17.0	1616.3	18.7
R109	1576.7	17.9	1548.5	16.5	1512.4	16.7	1531.1	19.8	1390.5	14.3	1397.2	14.9	1410.6	15.8	1512.4	20.5
R110	1707.8	17.9	1700.5	17.7	1754.3	26.0	1924.4	36.3	1562.3	17.9	1536.6	17.4	1683.4	26.4	1908.5	35.5
R111	1655.5	17.4	1588.8	18.1	1609.4	19.9	1865.3	36.1	1468.7	17.3	1518.5	16.1	1545.1	18.7	1852.0	35.1
R112	1734.4	16.0	1487.9	14.4	1410.3	14.5	1350.6	13.6	1236.9	13.9	1247.1	13.1	1256.1	12.8	1254.9	13.6
R100	1691.0	22.9	1582.9	21.6	1552.7	25.9	1664.7	31.9	1307.3	18.2	1328.5	19.0	1399.2	23.3	1536.5	28.9
C101	946.1	10.3	1020.5	17.6	1424.1	17.6	1556.4	24.0	924.6	10.3	1008.9	17.6	1422.4	17.6	1595.3	24.0
C102	1362.8	17.7	1468.2	19.4	1691.5	22.8	1795.9	23.2	1311.7	16.4	1441.1	18.7	1691.5	22.8	1768.7	23.2
C103	1618.0	14.6	1281.8	16.4	1683.1	18.1	1882.3	25.2	1602.3	14.6	1273.9	15.3	1664.4	18.1	1754.1	19.2
C104	1452.3	12.0	1357.0	13.9	1886.1	27.2	2032.2	24.5	1523.5	15.5	1318.1	14.9	1748.3	24.4	1911.1	23.8
C105	962.1	12.2	1071.5	13.1	1399.1	21.7	1516.8	26.7	888.0	10.3	1038.6	14.1	1361.2	20.7	1483.9	25.2
C106	1164.3	16.5	1267.4	21.2	1571.8	25.5	1702.7	30.4	1049.2	13.2	1171.2	20.3	1518.3	24.3	1576.8	25.8
C107	1350.0	10.2	1372.0	11.8	1738.0	15.4	1849.0	20.1	1199.0	10.0	1284.0	12.5	1668.0	15.6	1823.0	19.8
C108	1592.7	13.1	1538.7	13.7	1456.0	19.4	1524.9	22.4	1071.4	11.9	1119.0	13.7	1405.2	19.5	1493.1	20.1
C109	1348.7	19.1	1354.7	20.1	1153.7	15.8	1195.1	16.8	1082.0	13.0	1047.6	13.8	1093.1	15.5	1110.3	15.8
C100	1633.7	18.6	1630.4	20.0	1639.2	22.2	1703.0	26.1	1507.2	19.9	1560.0	20.0	1611.7	21.2	1703.0	25.8
RC101	2266.9	24.4	2170.0	21.8	2234.0	23.4	2488.1	24.2	2117.0	22.0	2122.5	22.0	2190.1	23.2	2488.1	24.0
RC102	2103.8	19.2	2023.0	19.2	2205.2	18.2	2380.5	23.0	1888.9	17.8	1921.6	17.6	2143.3	18.4	2380.5	22.6
RC103	1783.8	16.8	1738.9	17.8	1767.9	17.6	2056.2	25.2	1617.2	15.0	1637.5	15.0	1740.3	14.8	2054.8	21.8
RC104	1596.3	16.3	1612.8	14.7	1640.1	15.7	1610.5	22.3	1391.6	12.3	1398.7	12.0	1405.8	14.7	1506.3	17.0
RC105	2168.1	22.0	2187.4	22.0	2183.9	21.2	2261.0	23.8	1966.6	18.6	2024.4	18.8	2087.5	20.0	2157.6	22.2
RC106	2012.6	19.2	1918.6	18.4	1876.4	20.6	1932.7	20.2	1780.8	17.4	1787.1	17.6	1785.5	20.4	1909.2	20.8
RC107	1987.1	19.2	1896.2	17.0	1915.6	19.0	1933.4	17.4	1726.4	15.6	1726.4	15.0	1790.5	16.2	1939.4	16.4
RC108	1976.3	18.0	1750.3	16.2	1662.1	16.8	1575.2	16.6	1478.9	15.0	1499.0	14.8	1557.8	14.6	1512.3	14.8

Table A.7. Additional performance indicators of re-optimization strategies for MR2, MC2 and MRC2 instances and for 50% dod

Instance	FR								PR							
	SRR		NRR-1		NRR-2		NRR-3		SRR		NRR-1		NRR-2		NRR-3	
	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR	TD	NR
MR201	1525.4	5.0	1485.0	5.4	1449.9	6.0	1456.6	7.0	1389.2	5.0	1381.1	6.6	1389.2	7.6	1402.7	7.2
MR202	1481.7	5.8	1425.4	5.6	1389.6	6.6	1393.5	7.6	1297.6	5.2	1302.7	6.6	1327.0	6.4	1342.3	7.8
MR203	1353.7	5.2	1413.9	5.4	1489.8	6.6	1493.4	7.1	1293.5	5.0	1370.6	5.2	1480.2	6.4	1505.5	7.3
MR204	1383.6	6.2	1533.1	8.4	1476.0	7.8	1505.0	8.3	1216.5	4.5	1319.2	6.0	1431.3	7.2	1556.9	8.6
MR205	1286.7	6.0	1254.0	4.6	1197.2	5.2	1200.5	5.4	1144.8	4.6	1116.5	6.2	1121.9	6.0	1156.8	5.4
MR206	1324.1	6.3	1327.4	6.4	1217.3	4.8	1253.3	5.5	1148.7	4.5	1132.3	4.4	1130.1	4.4	1144.3	4.6
MR207	1388.0	5.2	1304.7	4.6	1240.7	4.4	1255.6	4.7	1212.2	4.2	1179.1	4.0	1209.9	4.4	1221.3	4.5
MR208	1365.5	6.2	1364.6	7.6	1352.1	7.2	1373.2	8.0	1164.8	4.4	1175.4	5.2	1330.0	6.4	1296.4	5.9
MR209	1295.9	6.6	1196.5	5.4	1187.8	5.6	1188.9	6.0	1157.6	4.2	1104.6	4.0	1110.0	4.1	1156.5	4.2
MR210	1257.5	4.2	1220.5	5.2	1161.3	5.4	1162.4	5.4	1092.6	4.2	1077.9	4.1	1105.3	4.6	1130.7	4.8
MR211	1244.7	4.0	1253.6	4.8	1167.9	4.0	1175.9	4.2	1105.1	4.2	1113.1	4.4	1116.1	4.6	1126.1	4.7
MC201	1945.5	6.8	1714.7	7.2	1651.3	8.0	1531.6	8.0	1476.7	8.0	1482.4	7.8	1478.1	7.8	1489.4	8.2
MC202	1371.4	7.8	1209.7	7.8	1185.6	8.2	1147.4	8.4	1163.8	8.2	1145.2	8.0	1134.3	8.0	1131.0	7.6
MC203	1469.9	6.8	1410.3	7.2	1348.9	7.6	1525.8	10.6	1188.5	6.8	1144.6	6.6	1311.3	7.4	1525.8	10.6
MC204	1290.7	7.0	1357.9	7.6	1618.5	9.0	1624.8	10.4	991.5	5.8	1038.1	6.0	1211.0	7.6	1274.6	10.2
MC205	1317.4	5.0	1063.9	5.7	1046.2	5.0	978.2	5.7	794.3	4.3	793.5	4.3	805.9	4.3	809.0	4.0
MC206	1039.3	5.2	914.7	4.6	843.3	5.4	822.6	5.4	724.9	4.4	731.9	4.2	756.1	4.8	738.1	4.6
MC207	1257.6	5.5	1061.5	5.0	986.4	5.0	1001.8	5.3	834.7	4.5	839.5	4.5	854.0	4.3	867.8	4.3
MC208	956.7	4.8	1024.6	4.6	865.9	5.2	822.3	5.4	755.8	5.2	746.5	5.0	735.7	4.8	727.2	5.0
MRC201	1897.7	6.0	1853.3	6.2	1799.5	7.0	1804.3	8.1	1644.4	6.0	1601.7	5.0	1603.3	5.1	1661.8	5.7
MRC202	1788.5	5.2	1655.6	5.8	1525.4	6.0	1576.1	7.5	1417.1	4.2	1389.7	5.4	1389.7	5.2	1439.0	6.2
MRC203	1862.5	5.4	1702.4	6.8	1605.3	9.4	1627.6	10.0	1497.6	4.6	1463.5	4.4	1600.0	5.8	1640.7	9.1
MRC204	1438.5	5.8	1507.9	7.4	1716.0	9.6	1728.8	10.2	1201.6	4.6	1361.7	5.2	1637.0	6.6	1707.5	9.2
MRC205	1780.7	6.8	1666.5	6.4	1650.8	7.0	1713.6	8.2	1500.8	5.6	1453.7	5.2	1469.4	5.4	1513.7	6.5
MRC206	1686.5	5.0	1541.9	5.4	1439.0	5.4	1471.6	6.3	1342.7	4.6	1325.7	4.8	1320.5	5.6	1367.4	6.0
MRC207	1592.8	5.0	1496.8	5.4	1405.7	5.8	1440.2	7.5	1371.2	5.6	1275.2	4.4	1291.2	4.4	1354.0	6.5
MRC208	1492.1	4.6	1390.0	4.6	1346.3	4.6	1401.3	5.7	1220.8	4.4	1204.0	4.0	1243.2	5.0	1255.5	5.2

Appendix B. Detailed results of the experiments presented in Chapter 6

We present here the detailed experimental results, which were summarized in Chapter 6. Specifically, Table B.1 presents the detailed results (in terms of VOI) of the re-optimization strategies for all instances and different values of fleet availability, when re-optimization depends on the number of DO received (Section 6.4.3). Tables B.2-B.4 present the detailed results of the re-optimization strategies for all instances, objectives and the different values of fleet availability in terms of VOI, under re-optimization cycles of known duration (Section 6.4.4).

Table B.1. Detailed performance of re-optimization strategies and different values of fleet availability

Instance	$V-0$								$V-2$								$V-4$							
	FR				PR				FR				PR				FR				PR			
	SRR	**NRR-1**	**NRR-2**	**NRR-3**	**SRR**	**NRR-1**	**NRR-2**	**NRR-3**	**SRR**	**NRR-1**	**NRR-2**	**NRR-3**	**SRR**	**NRR-1**	**NRR-2**	**NRR-3**	**SRR**	**NRR-1**	**NRR-2**	**NRR-3**	**SRR**	**NRR-1**	**NRR-2**	**NRR-3**
R101	5.3%	1.7%	2.2%	19.8%	3.5%	0.2%	2.2%	19.8%	10.8%	9.7%	10.9%	17.5%	8.0%	12.2%	10.9%	17.5%	9.6%	9.4%	11.7%	17.6%	7.2%	11.7%	11.7%	17.6%
R102	3.3%	9.7%	12.8%	17.6%	1.6%	6.5%	9.7%	14.3%	3.1%	4.6%	10.0%	11.3%	0.1%	2.9%	7.1%	11.3%	3.1%	5.7%	8.3%	9.7%	0.3%	2.9%	5.5%	8.2%
R103	24.5%	23.9%	24.2%	25.6%	11.9%	13.6%	14.6%	24.2%	11.5%	12.8%	16.8%	11.3%	5.7%	7.1%	5.8%	8.5%	6.0%	8.5%	8.4%	8.3%	4.3%	1.7%	5.6%	6.9%
R104	26.5%	19.4%	15.4%	22.9%	5.1%	2.6%	9.2%	11.7%	26.8%	25.3%	21.3%	22.5%	7.9%	9.3%	15.8%	15.9%	20.4%	17.9%	17.8%	12.9%	5.9%	7.1%	9.3%	11.7%
R105	9.8%	13.6%	9.1%	26.0%	3.7%	5.9%	9.1%	26.0%	12.1%	12.1%	13.2%	22.2%	5.2%	8.1%	13.2%	22.2%	16.6%	14.3%	19.7%	22.8%	8.7%	12.1%	18.5%	22.8%
R106	13.6%	9.9%	6.6%	13.8%	5.0%	5.0%	6.6%	8.9%	17.3%	17.2%	19.5%	25.4%	12.2%	17.1%	15.9%	24.1%	16.9%	16.7%	17.9%	26.6%	15.7%	18.9%	18.9%	26.4%
R107	30.9%	31.4%	23.3%	30.8%	3.8%	7.4%	9.5%	11.3%	24.7%	30.4%	24.3%	32.7%	10.2%	13.0%	14.2%	23.0%	13.4%	15.9%	10.6%	18.4%	3.9%	6.5%	6.5%	13.2%
R108	40.4%	40.4%	25.3%	27.0%	16.9%	16.8%	15.1%	16.8%	42.3%	35.9%	31.1%	32.7%	19.1%	16.3%	22.8%	22.4%	29.3%	24.4%	23.8%	15.6%	6.1%	9.7%	12.1%	10.7%
R109	13.0%	13.0%	5.3%	5.2%	3.3%	2.0%	3.9%	7.7%	9.4%	10.9%	7.3%	7.1%	0.1%	1.3%	3.5%	8.4%	6.5%	4.3%	5.2%	5.3%	1.0%	1.1%	2.1%	5.3%
R110	29.6%	15.3%	18.9%	24.4%	2.7%	2.7%	10.8%	19.9%	22.0%	13.4%	15.5%	19.9%	2.5%	6.7%	14.2%	18.6%	18.3%	22.4%	20.1%	27.6%	12.6%	14.7%	17.8%	24.3%
R111	24.9%	24.9%	20.0%	9.9%	2.6%	0.2%	9.8%	9.8%	19.0%	22.4%	20.3%	23.3%	11.1%	8.5%	13.9%	20.3%	13.5%	17.2%	21.9%	18.2%	7.4%	14.8%	15.8%	16.9%
R112	22.1%	19.9%	15.5%	23.5%	3.7%	3.7%	3.7%	9.6%	31.9%	32.5%	17.2%	23.1%	4.4%	5.4%	5.4%	14.5%	27.8%	21.4%	19.1%	26.4%	9.4%	13.6%	14.8%	23.0%
C101	3.0%	16.1%	33.1%	49.1%	3.0%	16.1%	33.1%	49.1%	1.1%	6.3%	29.3%	45.2%	1.1%	6.2%	28.3%	45.2%	0.2%	4.4%	25.4%	41.4%	0.2%	2.4%	24.4%	41.4%
C102	3.1%	9.9%	22.9%	63.0%	3.0%	9.9%	22.9%	63.0%	4.6%	12.3%	32.1%	62.8%	1.2%	12.3%	32.1%	62.8%	4.6%	11.6%	31.2%	61.1%	3.4%	11.6%	31.2%	61.1%
C103	13.0%	41.6%	53.8%	58.9%	12.7%	41.4%	53.8%	58.9%	35.3%	47.1%	58.9%	62.7%	35.3%	47.1%	58.9%	62.7%	39.7%	47.9%	54.7%	61.5%	38.3%	47.9%	54.7%	61.5%
C104	21.2%	22.2%	21.3%	10.5%	10.5%	10.5%	10.5%	10.4%	7.7%	7.9%	19.7%	16.9%	7.3%	7.5%	9.7%	7.7%	4.0%	7.5%	14.6%	14.4%	3.5%	7.2%	7.2%	7.2%
C105	4.1%	9.2%	28.0%	52.0%	3.9%	16.0%	28.0%	52.0%	1.3%	5.4%	24.3%	48.2%	0.0%	8.2%	22.2%	48.2%	0.4%	2.8%	20.6%	44.4%	0.1%	0.5%	18.5%	44.4%
C106	7.2%	11.2%	25.4%	39.4%	3.2%	8.1%	25.2%	39.4%	3.2%	4.3%	21.5%	35.4%	1.0%	3.2%	22.3%	35.4%	0.3%	2.3%	16.7%	31.7%	0.1%	0.2%	17.5%	31.7%
C107	54.2%	54.2%	60.5%	62.3%	54.2%	54.2%	60.5%	62.3%	37.8%	43.5%	55.6%	58.4%	33.6%	42.5%	55.6%	58.4%	22.9%	33.8%	49.7%	54.5%	20.9%	33.8%	49.7%	54.5%
C108	45.8%	46.6%	46.6%	45.6%	40.3%	43.3%	44.3%	45.6%	22.2%	20.9%	21.8%	29.6%	17.5%	19.5%	21.6%	29.6%	7.4%	6.1%	8.8%	23.7%	4.8%	4.7%	6.8%	23.7%
C109	63.3%	61.2%	63.2%	65.1%	60.1%	59.1%	62.2%	65.1%	31.1%	27.6%	30.6%	48.4%	25.5%	25.6%	27.4%	48.4%	5.1%	7.0%	13.8%	37.6%	3.7%	6.7%	13.7%	37.6%

Table B.2. Detailed performance of re-optimization strategies for the three objectives under vehicle availability V-0

Instance	$\breve{z}_1$								$\breve{z}_2$								$\breve{z}_3$							
	FR				**PR**				**FR**				**PR**				**FR**				**PR**			
	FTR1	**FTR2**	**FTR3**	**FTR4**	**FTR1**	**FTR2**	**FTR3**	**FTR4**	**FTR1**	**FTR2**	**FTR3**	**FTR4**	**FTR1**	**FTR2**	**FTR3**	**FTR4**	**FTR1**	**FTR2**	**FTR3**	**FTR4**	**FTR1**	**FTR2**	**FTR3**	**FTR4**
R101	1.7%	5.5%	17.2%	21.3%	1.6%	5.5%	17.2%	21.3%	1.7%	5.5%	17.2%	21.3%	1.7%	5.5%	17.2%	21.3%	3.7%	5.5%	17.2%	21.3%	3.7%	5.5%	17.2%	21.3%
R102	10.3%	11.4%	12.6%	13.7%	12.6%	11.5%	12.7%	12.6%	10.5%	13.7%	14.9%	11.8%	10.6%	14.8%	17.2%	11.8%	13.8%	14.9%	13.8%	14.0%	11.7%	14.9%	13.8%	12.9%
R103	30.6%	25.6%	26.0%	34.1%	16.9%	21.7%	25.0%	28.1%	22.9%	24.9%	27.9%	30.9%	18.1%	18.7%	24.0%	27.0%	21.7%	22.6%	23.2%	28.8%	19.9%	19.8%	22.9%	26.8%
R104	27.0%	21.3%	25.7%	22.9%	6.1%	14.4%	17.3%	21.3%	24.3%	16.2%	25.8%	24.4%	7.5%	6.1%	15.9%	21.4%	20.1%	21.5%	24.3%	24.2%	2.8%	6.0%	10.3%	18.7%
R105	5.2%	12.1%	22.5%	29.3%	5.2%	12.1%	22.5%	29.3%	3.9%	12.1%	22.5%	31.2%	5.3%	12.1%	22.5%	31.2%	5.9%	10.2%	22.5%	29.7%	5.9%	10.2%	22.5%	29.7%
R106	11.8%	10.5%	16.2%	16.3%	6.1%	5.8%	10.3%	16.2%	13.0%	14.4%	15.1%	9.5%	8.4%	10.7%	10.4%	9.3%	7.3%	10.7%	12.9%	10.1%	8.4%	12.0%	12.8%	13.8%
R107	30.4%	31.6%	27.7%	25.5%	9.5%	15.6%	16.7%	19.0%	24.7%	29.2%	26.3%	25.6%	6.2%	10.7%	17.9%	17.7%	19.6%	27.9%	26.8%	19.8%	0.3%	7.2%	13.2%	14.6%
R108	37.6%	28.3%	34.9%	26.6%	18.5%	18.7%	22.4%	21.0%	38.5%	28.7%	35.0%	28.0%	16.1%	19.1%	21.2%	20.0%	26.1%	22.1%	31.8%	23.2%	2.6%	5.8%	18.8%	15.1%
R109	12.8%	9.7%	13.4%	2.2%	7.7%	10.0%	6.7%	3.2%	12.8%	10.5%	13.5%	2.2%	6.5%	9.7%	7.6%	3.2%	12.2%	9.4%	12.7%	3.9%	8.5%	8.7%	8.5%	4.0%
R110	26.3%	23.2%	25.0%	22.1%	4.3%	11.2%	17.8%	14.0%	23.8%	19.6%	26.2%	21.5%	4.3%	10.3%	17.8%	10.5%	19.0%	19.6%	13.5%	13.5%	4.9%	6.2%	8.0%	15.9%
R111	24.2%	19.4%	23.8%	23.9%	6.4%	9.4%	17.5%	15.8%	22.7%	18.0%	23.8%	23.9%	8.1%	4.8%	19.2%	17.3%	12.9%	24.2%	23.8%	20.7%	10.0%	5.0%	12.7%	19.2%
R112	24.0%	16.1%	15.1%	23.6%	3.9%	8.0%	8.5%	15.6%	24.0%	16.2%	16.5%	25.0%	5.3%	9.4%	14.7%	13.2%	29.2%	24.4%	27.1%	24.0%	6.4%	8.1%	5.3%	16.0%
R100	39.4%	42.8%	39.3%	37.3%	18.9%	29.1%	27.4%	27.3%	39.4%	41.1%	42.7%	37.3%	30.5%	25.7%	28.7%	25.7%	44.4%	42.8%	42.7%	35.6%	28.8%	22.3%	24.0%	25.7%

Table B.3. Detailed performance of re-optimization strategies for the three objectives under vehicle availability V-2

Instance	$\breve{z}_1$								$\breve{z}_2$								$\breve{z}_3$							
	FR				**PR**				**FR**				**PR**				**FR**				**PR**			
	FTR1	**FTR2**	**FTR3**	**FTR4**	**FTR1**	**FTR2**	**FTR3**	**FTR4**	**FTR1**	**FTR2**	**FTR3**	**FTR4**	**FTR1**	**FTR2**	**FTR3**	**FTR4**	**FTR1**	**FTR2**	**FTR3**	**FTR4**	**FTR1**	**FTR2**	**FTR3**	**FTR4**
R101	8.7%	12.3%	18.4%	24.1%	8.7%	12.3%	18.4%	24.1%	8.7%	10.4%	18.4%	24.1%	8.7%	10.4%	18.4%	24.1%	11.2%	10.4%	18.4%	24.1%	11.2%	10.4%	18.4%	24.1%
R102	6.6%	8.4%	13.2%	10.4%	8.5%	10.3%	12.3%	11.4%	9.6%	10.4%	15.2%	10.6%	10.5%	12.2%	15.2%	10.6%	11.4%	12.2%	13.3%	10.6%	11.4%	12.2%	13.3%	11.6%
R103	14.5%	12.8%	7.8%	15.3%	6.1%	8.6%	8.7%	12.3%	8.3%	7.3%	12.7%	8.9%	10.0%	8.2%	10.7%	12.9%	13.0%	9.2%	15.7%	12.0%	9.2%	9.2%	13.6%	12.9%
R104	27.0%	27.8%	28.7%	25.9%	12.1%	18.1%	19.0%	22.4%	21.9%	17.6%	24.3%	21.6%	8.8%	8.0%	16.5%	19.0%	16.8%	16.7%	21.9%	23.3%	7.0%	12.2%	15.7%	19.9%
R105	10.6%	10.3%	19.1%	25.6%	11.5%	10.3%	19.1%	25.6%	10.6%	11.2%	19.4%	28.2%	10.6%	11.2%	19.4%	28.2%	10.8%	11.1%	19.2%	27.3%	8.7%	11.1%	19.2%	27.3%
R106	21.0%	21.8%	27.4%	25.6%	18.5%	20.0%	21.7%	28.0%	17.1%	19.4%	24.4%	23.4%	21.1%	23.5%	21.9%	24.3%	22.8%	20.2%	21.8%	25.0%	21.2%	21.9%	22.6%	26.7%
R107	27.2%	29.7%	26.6%	29.9%	19.3%	17.0%	23.5%	28.0%	25.8%	23.9%	24.8%	26.4%	14.0%	13.3%	21.4%	26.4%	24.5%	23.5%	25.6%	27.3%	15.3%	18.5%	23.4%	26.6%
R108	34.7%	34.8%	28.8%	28.2%	16.6%	19.0%	19.6%	21.1%	35.4%	27.0%	24.2%	24.5%	15.6%	15.7%	17.6%	18.1%	25.0%	26.2%	22.6%	24.5%	12.1%	12.0%	15.8%	17.1%
R109	11.9%	12.3%	8.9%	7.1%	6.2%	7.0%	6.6%	7.0%	11.9%	10.7%	9.3%	6.3%	6.3%	6.9%	3.1%	6.2%	13.2%	12.3%	7.9%	6.3%	5.7%	5.4%	5.4%	6.2%
R110	21.4%	15.9%	17.3%	15.3%	9.6%	13.4%	14.8%	19.6%	21.2%	14.4%	22.4%	17.4%	9.5%	12.4%	15.0%	16.3%	12.6%	14.6%	15.2%	15.3%	8.8%	12.8%	12.1%	16.4%
R111	19.7%	22.7%	24.9%	22.5%	10.6%	14.5%	17.4%	17.3%	21.1%	21.1%	27.0%	22.5%	10.3%	12.6%	20.4%	20.2%	18.9%	21.2%	28.5%	24.5%	6.8%	13.6%	21.0%	19.2%
R112	28.2%	22.2%	22.8%	21.3%	5.4%	8.3%	12.8%	16.5%	25.7%	21.2%	23.7%	29.1%	5.3%	6.6%	14.7%	18.3%	18.0%	19.6%	26.3%	23.9%	3.1%	10.2%	13.8%	17.6%
R100	36.8%	34.9%	31.8%	37.4%	20.4%	25.2%	28.8%	34.4%	37.7%	33.7%	35.7%	32.7%	18.8%	22.2%	21.4%	28.4%	28.3%	35.3%	34.1%	36.3%	16.7%	20.9%	20.9%	28.3%

Table B.4. Detailed performance of re-optimization strategies for the three objectives under vehicle availability V-4

Instance	$\check{z}_1$								$\check{z}_2$								$\check{z}_3$							
	FR				PR				FR				PR				FR				PR			
	FTR1	FTR2	FTR3	FTR4	FTR1	FTR2	FTR3	FTR4	FTR1	FTR2	FTR3	FTR4	FTR1	FTR2	FTR3	FTR4	FTR1	FTR2	FTR3	FTR4	FTR1	FTR2	FTR3	FTR4
R101	8.0%	12.5%	19.5%	22.0%	8.0%	12.5%	19.5%	22.0%	8.0%	11.0%	19.5%	22.0%	8.0%	11.0%	19.5%	22.0%	8.8%	11.9%	19.5%	22.0%	8.8%	11.9%	19.5%	22.0%
R102	6.1%	7.7%	13.7%	10.3%	7.8%	9.5%	14.5%	10.2%	5.3%	8.6%	15.5%	11.3%	7.1%	8.6%	15.5%	11.3%	6.3%	8.6%	13.9%	10.5%	6.3%	8.6%	13.9%	10.5%
R103	8.4%	9.3%	8.9%	8.9%	7.3%	8.9%	8.0%	9.8%	5.0%	5.1%	7.5%	9.2%	8.6%	5.0%	9.2%	9.3%	4.2%	3.2%	7.4%	10.1%	5.9%	4.1%	7.5%	9.2%
R104	18.1%	15.7%	19.5%	16.2%	6.3%	8.6%	11.7%	13.1%	11.3%	8.8%	8.8%	12.7%	7.3%	7.2%	8.7%	12.6%	7.5%	6.6%	8.8%	11.1%	5.7%	4.2%	8.7%	12.6%
R105	14.5%	15.2%	20.0%	27.2%	15.2%	14.5%	20.0%	27.2%	13.8%	13.9%	19.3%	27.3%	14.0%	13.9%	19.3%	27.3%	13.8%	13.1%	18.5%	26.5%	11.5%	13.1%	18.5%	26.5%
R106	17.5%	18.2%	26.2%	21.0%	18.2%	18.8%	26.2%	23.2%	17.6%	18.9%	19.8%	21.2%	19.1%	20.4%	21.9%	21.2%	15.5%	21.1%	19.7%	23.2%	14.8%	21.2%	20.5%	23.2%
R107	15.8%	18.1%	19.1%	25.8%	14.8%	15.4%	16.4%	21.5%	19.3%	16.7%	21.1%	21.0%	14.0%	10.5%	16.6%	17.4%	17.8%	15.0%	20.0%	22.8%	12.3%	8.7%	15.7%	20.0%
R108	20.1%	15.1%	11.1%	12.1%	3.3%	7.3%	9.5%	9.6%	17.2%	8.3%	11.3%	12.0%	5.0%	6.5%	9.6%	11.3%	5.9%	9.0%	10.5%	12.8%	4.2%	4.0%	8.0%	12.0%
R109	3.7%	7.0%	7.3%	4.2%	2.2%	4.3%	3.6%	3.5%	3.0%	7.1%	5.7%	5.8%	1.6%	3.0%	3.0%	5.8%	4.4%	4.3%	5.7%	5.8%	0.9%	2.9%	3.5%	5.8%
R110	23.2%	24.0%	17.4%	24.7%	15.0%	16.4%	16.6%	23.2%	21.8%	16.5%	18.1%	25.6%	14.3%	15.8%	17.3%	25.6%	21.8%	17.5%	18.2%	21.2%	15.3%	16.0%	18.8%	22.6%
R111	21.2%	22.0%	22.1%	22.9%	14.2%	16.3%	18.9%	21.2%	19.8%	17.6%	20.6%	25.9%	14.8%	14.1%	19.7%	27.2%	17.6%	16.0%	20.4%	26.6%	14.4%	15.8%	19.6%	24.3%
R112	24.0%	20.3%	24.9%	27.8%	9.1%	14.7%	18.8%	23.7%	17.8%	15.0%	23.9%	17.0%	8.5%	8.0%	16.8%	14.1%	14.3%	14.9%	16.2%	17.6%	6.5%	6.5%	14.8%	16.3%
R100	34.1%	36.5%	36.3%	37.3%	25.2%	28.4%	33.9%	34.2%	32.1%	27.2%	17.5%	24.5%	23.7%	23.9%	21.3%	28.7%	22.4%	22.5%	21.4%	28.7%	20.1%	21.6%	20.6%	29.5%