



University of the Aegean

Department of Financial & Management Engineering

# **THE MULTI-PERIOD VEHICLE ROUTING PROBLEM AND ITS APPLICATIONS**

**THEODORE ATHANASOPOULOS**

Chios, Greece

May 2011



---

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

**Ph.D. Dissertation**

**BY THEODORE ATHANASOPOULOS**

**THE MULTI-PERIOD VEHICLE ROUTING PROBLEM**  
**AND ITS APPLICATIONS**

**ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ**

**ΙΩΑΝΝΗΣ ΜΙΝΗΣ**, ΚΑΘΗΓΗΤΗΣ (Επιβλέπων)

Τμήμα Μηχανικών Οικονομίας και Διοίκησης του Πανεπιστημίου Αιγαίου

**ΓΕΩΡΓΙΟΣ ΔΟΥΝΙΑΣ**, ΚΑΘΗΓΗΤΗΣ (Μέλος συμβουλευτικής επιτροπής, σε εκπαιδευτική άδεια)

Τμήμα Μηχανικών Οικονομίας και Διοίκησης του Πανεπιστημίου Αιγαίου

**ΕΠΑΜΕΙΝΩΝΔΑΣ ΚΥΡΙΑΚΙΔΗΣ**, ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ (Μέλος συμβουλευτικής επιτροπής)

Τμήμα Στατιστικής του Οικονομικού Πανεπιστημίου Αθηνών

**ΝΙΚΟΛΑΟΣ ΑΜΠΑΖΗΣ**, ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ (Μέλος εξεταστικής επιτροπής)

Τμήμα Μηχανικών Οικονομίας και Διοίκησης του Πανεπιστημίου Αιγαίου

**ΔΗΜΟΣΘΕΝΗΣ ΔΡΙΒΑΛΙΑΡΗΣ**, ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ (Μέλος εξεταστικής επιτροπής)

Τμήμα Μηχανικών Οικονομίας και Διοίκησης του Πανεπιστημίου Αιγαίου

**ΓΕΩΡΓΙΟΣ ΙΩΑΝΝΟΥ**, ΚΑΘΗΓΗΤΗΣ (Μέλος εξεταστικής επιτροπής)

Τμήμα Διοικητικής Επιστήμης & Τεχνολογίας του Οικονομικού Πανεπιστημίου Αθηνών

**ΛΑΜΠΡΟΣ ΛΑΪΟΣ**, ΚΑΘΗΓΗΤΗΣ (Μέλος εξεταστικής επιτροπής)

Τμήμα Βιομηχανικής Διοίκησης & Τεχνολογίας του Πανεπιστημίου Πειραιά

**ΑΓΑΠΙΟΣ ΠΛΑΤΗΣ**, ΚΑΘΗΓΗΤΗΣ (Μέλος εξεταστικής επιτροπής)

Τμήμα Μηχανικών Οικονομίας και Διοίκησης του Πανεπιστημίου Αιγαίου

**ΧΙΟΣ, 2011**

---



## ACKNOWLEDGEMENTS

*Firstly and most importantly, I would like to thank Professor Ioannis Minis for his valuable supervision and guidance throughout these years, for his strong belief in me, and for his substantial contribution to my academic and professional evolvement. Secondly, I gratefully thank my close friend and colleague Ninikas Georgios, PhD Candidate of the University of the Aegean, for his endless support and encouragement, as well as for the countless discussions through these PhD years.*

*Furthermore, I am grateful to George Dikas, PhD Candidate of the University of the Aegean, for his ideas and support. I would also like to acknowledge the support and the resources made available to me through the DeOPSys Lab of the Financial and Management Engineering (FME) Department of the University of the Aegean.*

*I would also like to thank my family for being there and supporting me at every decision I make. Additionally, I would like to thank my partner Maria Sourmpati for supporting me in starting my PhD adventure and for her endless support and patience throughout all these years. Finally, I would like to mention my friend and fellow student, Nikolaos Spyridon Stogiannos, for supporting and encouraging my research.*

*to my family*





This research is part of the 03ED067 project, implemented within the framework of the “Reinforcement Programme of Human Research Manpower” (PENED) and co-financed by National and Community Funds (20% from the Greek Ministry of Development – General Secretariat of Research and Technology and 80% from E.U.-European Social Fund).

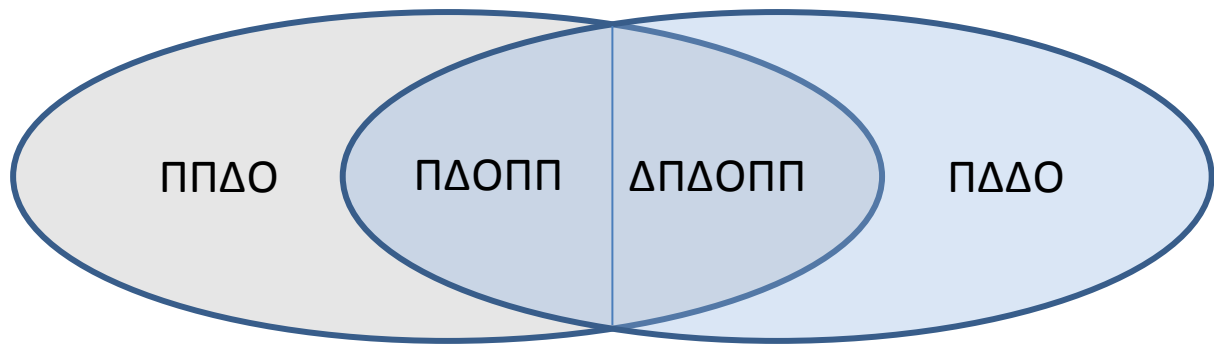


## SUMMARY (IN GREEK)

Στην παρούσα διδακτορική διατριβή διερευνάται το Πρόβλημα Δρομολόγησης Οχημάτων Πολλαπλών Περιόδων (ΠΔΟΠΠ) με Χρονικά Παράθυρα (ΠΔΟΠΠΧΠ). Στόχος του προβλήματος αυτού είναι η ελαχιστοποίηση του κόστους δρομολόγησης εντός ορίζοντα προγραμματισμού πολλαπλών περιόδων (π.χ. ημερών), λαμβάνοντας υπόψη περιορισμούς χρονικών παραθύρων και χωρητικότητας οχημάτων, καθώς και χρονικών παραθύρων περιόδων. Ως χρονικό παράθυρο περιόδων πελάτη ορίζεται σύνολο συνεχών περιόδων εντός των οποίων ο πελάτης επιθυμεί να εξυπηρετηθεί. Το Σχήμα Π.1, παρουσιάζει τα ΠΔΟΠΠ τα οποία μελετώνται στην παρούσα διατριβή και τα οποία εμπίπτουν σε δύο κατηγορίες:

- Η πρώτη κατηγορία περιλαμβάνει το βασικό ΠΔΟΠΠ, στο οποίο όλοι οι πελάτες εντός του ορίζοντα προγραμματισμού των επόμενων  $P$  περιόδων θεωρούνται γνωστοί. Το πρόβλημα αυτό μπορεί να θεωρηθεί ως μία ειδική περίπτωση του Προβλήματος Περιοδικής Δρομολόγησης Οχημάτων (ΠΠΔΟ). Στο τελευταίο, οι πελάτες εξυπηρετούνται πολλαπλές φορές εντός χρονικού ορίζοντα προγραμματισμού  $P$  περιόδων σύμφωνα με επιθυμητό πλάνο εξυπηρέτησης (π.χ. Δευτέρα – Τετάρτη – Παρασκευή).
- Η δεύτερη κατηγορία αφορά το ΠΔΟΠΠ Εκτεταμένου Χρονικού Ορίζοντα (ΠΔΟΠΠΕΧΟ), π.χ.  $S$  περιόδων.. Το πρόβλημα αυτό αντιμετωπίζεται μέσω προγραμματισμού κυλιόμενου χρονικού ορίζοντα. Σύμφωνα με αυτή την τεχνική, σε κάθε περίοδο επιλύεται ένα ΠΔΟΠΠ για  $P < S$  περιόδους (χωρίς να είναι γνωστό το σύνολο των αιτημάτων εξυπηρέτησης εντός του ορίζοντα προγραμματισμού των  $P$  περιόδων). Με βάση τη λύση, υλοποιείται η πρώτη περίοδος του ΠΔΟΠΠ και η διαδικασία επαναλαμβάνεται για κάθε επόμενη περίοδο του ορίζοντα  $S$ . Μελετώνται δύο διαφορετικές περιπτώσεις: (α) η ημι-στατική περίπτωση, στην οποία όλα τα αιτήματα εντός του χρονικού ορίζοντα  $S$  θεωρούνται γνωστά, και (β) η δυναμική περίπτωση όπου νέα αιτήματα εμφανίζονται σε κάθε περίοδο του ορίζοντα προγραμματισμού.

Μία παραλλαγή της δυναμικής περίπτωσης του ΠΔΟΠΠΕΧΟ (ΔΠΔΟΠΠΕΧΟ) έχει μελετηθεί από τους Angelelli *et al.* (2009) και Wen *et al.* (2009) και αφορά την επιπρόσθετη δυνατότητα τροποποίησης των εκτελούμενων δρομολογίων (δηλ. οχημάτων που βρίσκονται καθοδόν) για να εξυπηρετηθούν νέα δυναμικά αιτήματα. Το πρόβλημα αυτό μπορεί να θεωρηθεί ότι ανήκει στη κατηγορία των Προβλημάτων Δυναμικών Δρομολόγησης Οχημάτων (ΠΔΔΟ).



Σχήμα Π.1: Προβλήματα που σχετίζονται με περιβάλλοντα πολλαπλών περιόδων

Η υφιστάμενη βιβλιογραφία στο τομέα των ΠΔΟΠΠ είναι περιορισμένη, όπως έχει τονιστεί και από τους Bostel *et al.* (2008) και Wen *et al.* (2009). Ο Πίνακας Π.1 παρουσιάζει τα βασικά χαρακτηριστικά των προβλημάτων που έχουν διερευνηθεί στις σχετικές δημοσιεύσεις.

Table Π.1: Προβλήματα Δρομολόγησης Οχημάτων Πολλαπλών Περιόδων

	Πολλαπλά Οχήματα	Χρονικά Παράθυρα	Παράθυρα Περιοδών (# Περιόδων)	Μέθοδος Επίλυσης	Σταθερά Δρομολόγησης
Teng <i>et al.</i> (2006)				Ευρετική/ Δυναμική Δημιουργία Μεταβλητών	
Angelelli <i>et al.</i> (2007)			1 ή 2	Ευρετική	
Andreatta and Lulli (2008)			1 ή 2	Markov Process	
Tricoire (2006; 2007), Bostel <i>et al.</i> (2008)	✓	✓	1 ή 2	Μετευρετική/ Δυναμική Δημιουργία Μεταβλητών	
Wen <i>et al.</i> (2010)	✓		1 έως 15 (2.5 κ.μ.ο.)	Ευρετική	
Angelelli <i>et al.</i> (2009)	✓		1 ή 2	Ευρετική	
Athanasopoulos and Minis (2010)	✓	✓	5	Ευρετική	✓

Για την αντιμετώπιση του ΠΔΟΠΠΧΠ προτείνουμε προσέγγιση ακριβούς επίλυσης (exact) που χρησιμοποιεί τη μέθοδο Δυναμικής Δημιουργίας Μεταβλητών (ΔΔΜ) ή Column Generation (CG). Προτείνουμε δύο καινοτόμες τεχνικές για την επιτάχυνση της εύρεσης κατώτατων ορίων της λύσης (lower bounds), δηλαδή για την επίλυση της γραμμικής χαλάρωσης του ΠΔΟΠΠΧΠ. Οι τεχνικές αυτές εκμεταλλεύονται το περιβάλλον πολλαπλών περιόδων του προβλήματος, και χρησιμοποιούν τις ομοιότητες μεταξύ των διαφορετικών υποπροβλημάτων της μεθόδου ΔΔΜ. Για την εύρεση ακέραιων λύσεων χρησιμοποιούμε δύο διαφορετικές στρατηγικές branch-and-price οι οποίες λαμβάνουν υπόψη τα χαρακτηριστικά των πολλαπλών περιόδων. Προτείνουμε, επίσης απλή ευρετική μέθοδο, η οποία επιταχύνει επιπλέον την διαδικασία επίλυσης με αμελητέα διαφοροποίηση του κόστους της λύσης από το βέλτιστο κόστος.

Όσον αφορά το ΠΔΟΠΠΧΠ Εκτεταμένου Χρονικού Ορίζοντα (ΠΔΟΠΠΧΠΕΧΟ), διερευνούμε την χρήση της τεχνικής προγραμματισμού κυλιόμενου χρονικού ορίζοντα. Αρχικά προτείνονται τρία θεωρήματα τα οποία παρέχουν σημαντικές πληροφορίες σχετικά με τις βασικές παραμέτρους της διαδικασίας επίλυσης: Τον ορίζοντα προγραμματισμού και τον ορίζοντα υλοποίησης. Επιπρόσθετα, μελετώνται σημαντικές τροποποιήσεις του ΠΔΟΠΠΧΠ και των προτεινόμενων μεθόδων επίλυσής του μέσω των οποίων καθίσταται δυνατή η μετάθεση της εξυπηρέτησης πελατών σε επιτρεπτές περιόδους πέραν του ορίζοντα προγραμματισμού. Χρησιμοποιώντας τις τροποποιημένες προσεγγίσεις, μελετάται η επίλυση της ημι-στατικής και το δυναμικής περίπτωσης (που αναφέρθηκαν παραπάνω) λαμβάνοντας υπόψη διαφορετικές συνθήκες, όπως η γεωγραφική κατανομή των πελατών και το εύρος των χρονικών παραθύρων. Η ανάλυση διερευνά και προτείνει τις κατάλληλες τιμές του ορίζοντα προγραμματισμού και του ορίζοντα υλοποίησης για τις διάφορες περιπτώσεις. .

Τελικά, μελετάται παραλλαγή του προβλήματος ΠΔΟΠΠΧΠΕΧΟ που παρουσιάζει σημαντικό πρακτικό ενδιαφέρον. Η περίπτωση αυτή αφορά σε υβριδικό μοντέλο εξυπηρέτησης των πελατών και περιλαμβάνει (α) μη-ευέλικτες και (β) ευέλικτες παραγγελίες πελατών. Προτείνονται οι απαραίτητες τροποποιήσεις του μαθηματικού μοντέλου καθώς και της μεθόδου επίλυσης και μελετάται η επίλυση του προβλήματος υπό διαφορετικές συνθήκες. Όπως και προηγουμένως η πειραματική ανάλυση εντοπίζει τις κατάλληλες τιμές του ορίζοντα προγραμματισμού και του ορίζοντα υλοποίησης για τις διάφορες περιπτώσεις.

## ΜΑΘΗΜΑΤΙΚΟ ΜΟΝΤΕΛΟ ΤΟΥ ΠΔΟΠΠΧΠ

Δίνεται ορίζοντας προγραμματισμού  $|P|$  περιόδων και ορίζεται ως  $p_c = 0$  η τρέχουσα περίοδος. Θεωρούμε ότι όλοι οι πελάτες πρέπει να εξυπηρετηθούν εντός των επόμενων  $P$  περιόδων, δηλαδή εντός του ορίζοντα προγραμματισμού  $[p_c + 1, p_c + P]$ . Οι παράμετροι του μαθηματικού μοντέλου είναι οι εξής:

$H$	Σύνολο των $P$ συνεχόμενων περιόδων (ορίζοντας προγραμματισμού)
$N = \{1, \dots, n\}$	Γνωστοί πελάτες (ή παραγγελίες) κατά την έναρξη της περιόδου 1
$W = N \cup \{0, n + 1\}$	Σύνολο κόμβων, συμπεριλαμβανομένων της αρχικής και τελικής αποθήκης (depot). Κάθε όχημα εκκινεί από την αρχική αποθήκη (κόμβος 0) και τερματίζει στην τελική αποθήκη (κόμβος $n + 1$ ). Επισημαίνεται ότι για την αποθήκη χρησιμοποιούνται δύο διαφορετικοί κόμβοι ώστε να επιτρέπεται κάποιο όχημα να μείνει ανενεργό (Cordeau <i>et al.</i> , 2002)

$A = \{(i, j) : i, j \in W\}$	Σύνολο όλων των δυνατών ακμών ανάμεσα στους κόμβους του συνόλου $W$
$I_i = [\xi_i^s, \xi_i^e]$	Χρονικό παράθυρο περιόδων του πελάτη $i$ ; όπου $1 \leq \xi_i^s \leq \xi_i^e \leq P$
$c_{ij}$	Κόστος διάνυσης της ακμής $(i, j), \{i, j \in W\}$
$t_{ij}$	Χρόνος διάνυσης της ακμής $(i, j), \{i, j \in W\}$ , συμπεριλαμβανομένου του χρόνου εξυπηρέτησης του πελάτη $i$
$d_i$	Ζήτηση του πελάτη $i, \{i \in N\}$
$K_p$	Σύνολο των $ K_p $ διαθέσιμων οχημάτων ανά περίοδο $p, \{p \in H\}$
$Q_k^p$	Χωρητικότητα του οχήματος $k$ κατά την περίοδο $p$
$[a_i, b_i]$	Χρονικό παράθυρο εξυπηρέτησης του πελάτη $i$ , κοινό για κάθε περίοδο εντός του παραθύρου περιόδων $I_i$ ; Για τους κόμβους 0 και $n + 1$ , ο νωρίτερος χρόνος εκκίνησης κάθε οχήματος από το αμαξοστάσιο ορίζεται ως $a_0 = a_{n+1}$ , ενώ ο αργότερος χρόνος επιστροφής κάθε οχήματος στην αποθήκη, ορίζεται ως $b_0 = b_{n+1}$ .

Ορίζονται δύο διαφορετικά σύνολα μεταβλητών: (α) Η μεταβλητή  $x_{ijpk}$  είναι ίση με ένα (1) εάν το όχημα  $k$  διανύει την ακμή  $(i, j)$  εντός της περιόδου  $p$  και μηδέν σε οιαδήποτε άλλη περίπτωση. (β) Η μεταβλητή  $s_{ipk}$  αφορά στο χρόνο έναρξης εξυπηρέτησης του πελάτη (κόμβου)  $i$  από το όχημα  $k$  εντός της περιόδου  $p$ . Επισημαίνεται ότι το  $s_{ipk}$  ισούται με μηδέν (0) εάν ο κόμβος  $i$  δεν εξυπηρετείται από το όχημα  $k$  εντός της περιόδου  $p$ .

Αντικειμενικός στόχος του προβλήματος είναι η ελαχιστοποίηση του συνολικού κόστους δρομολόγησης καθ' όλο το εύρος του ορίζοντα προγραμματισμού και δίνεται από την εξίσωση:

$$\min(z) = \sum_{p \in H} \sum_{k \in K_p} \sum_{(i,j) \in A} c_{ij} x_{ijpk} \quad (\text{Π.1})$$

Υπό τους περιορισμούς

$$\sum_{p \in I_i} \sum_{k \in K_p} \sum_{j \in N \cup \{n+1\}} x_{ijpk} = 1 \quad \forall i \in N \quad (\text{Π.2})$$

$$\sum_{p \notin I_i} \sum_{k \in K_p} \sum_{j \in N \cup \{n+1\}} x_{ijpk} = 0 \quad \forall i \in N \quad (\text{Π.3})$$

$$\sum_{j \in N \cup \{n+1\}} x_{0jpk} = 1 \quad \forall p \in H, \forall k \in K_p \quad (\text{Π.4})$$

$$\sum_{i \in N \cup \{0\}} x_{ijpk} - \sum_{i' \in N \cup \{n+1\}} x_{ji'pk} = 0 \quad \forall p \in H, \forall k \in K_p, \forall j \in N \quad (\text{Π.5})$$

$$\sum_{j \in N \cup \{0\}} x_{j,n+1,pk} = 1 \quad \forall p \in H, \forall k \in K_p \quad (\text{Π.6})$$

$$\sum_{i \in N} d_i \sum_{j \in N \cup \{n+1\}} x_{ijpk} \leq Q_k^p \quad \forall p \in H, \forall k \in K_p \quad (\text{Π.7})$$

$$s_{ipk} + t_{ij} - M(1 - x_{ijpk}) \leq s_{jpk} \quad \forall p \in H, \forall k \in K_p, \forall (i, j) \in A \quad (\text{Π.8})$$

$$a_i \sum_{j \in N \cup \{n+1\}} x_{ijpk} \leq s_{ipk} \leq b_i \sum_{j \in N \cup \{n+1\}} x_{ijpk} \quad \forall p \in H, \forall k \in K_p, \forall i \in N \quad (\text{Π.9})$$

$$a_i \leq s_{ipk} \leq b_i \quad \forall p \in H, \forall k \in K_p, i \in \{0, n+1\} \quad (\text{Π.10})$$

$$x_{ijpk} \in \{0, 1\} \quad \forall p \in H, \forall k \in K_p, (i, j) \in A \quad (\text{Π.11})$$

Η αντικειμενική συνάρτηση (Π.1) αφορά στο συνολικό κόστος δρομολόγησης. Μέσω των περιορισμών (Π.2) και (Π.3) κάθε πελάτης πρέπει να εξυπηρετηθεί μόνο μία φορά (από ένα όχημα και εντός μίας περιόδου) εντός του αντίστοιχου παραθύρου περιόδων. Οι περιορισμοί (Π.4) και (Π.6) ορίζουν ότι κάθε όχημα αναχωρεί και τερματίζει στην αρχική και τελική αποθήκη, αντίστοιχα. Οι περιορισμοί (Π.5) αναφέρονται στη διατήρηση της ροής κάθε οχήματος. Μέσω των περιορισμών (Π.7) καθορίζεται ότι το φορτίο κάθε οχήματος δε θα ξεπεράσει τη χωρητικότητά του. Οι περιορισμοί (Π.8) και (Π.9) καθορίζουν ότι κάθε πελάτης εξυπηρετείται εντός του χρονικού παραθύρου του, ενώ οι περιορισμοί (Π.10) αφορούν στο χρονικό παράθυρο της αποθήκης. Επισημαίνεται ότι το  $M$  συμβολίζει ένα μεγάλο θετικό αριθμό. Τέλος, οι περιορισμοί (Π.11) δεσμεύουν τις μεταβλητές ροής σε δυαδικές τιμές  $\{0, 1\}$ .

Επιλύουμε της γραμμική «χαλάρωση» του ανωτέρω προβλήματος μέσω της μεθόδου ΔΔΜ για την εύρεση κατώτατων ορίων (lower bounds). Η ΔΔΜ διασπά (decomposes) το χαλαρωμένο μοντέλο σε ένα Κυρίως Πρόβλημα (ΚΠ) και πολλαπλά Υποπροβλήματα (ΥΠ). Για την εύρεση ακέραιων λύσεων χρησιμοποιείται η μέθοδος branch-and-price, κατά την οποία η διαδικασία ΔΔΜ χρησιμοποιείται σε κάθε κόμβο του σχετικού δένδρου.

## ΔΙΑΣΠΑΣΗ ΤΟΥ ΜΑΘΗΜΑΤΙΚΟΥ ΜΟΝΤΕΛΟΥ

Στην παρούσα ενότητα παρουσιάζουμε την διάσπαση του μαθηματικού μοντέλου για τη γενική περίπτωση του ΠΔΟΠΠΧΠ. Στην περίπτωση αυτή, το ΚΠ περιλαμβάνει μόνο τους περιορίσιμους αναφορικά με το μέγεθος του στόλου οχημάτων και τους *σύνθετους* (complex) περιορισμούς (που αφορούν όλες τις περιόδους συνδυαστικά). Τα ΥΠ περιλαμβάνουν τους λοιπούς περιορισμούς αναφορικά με την εφικτότητα των δρομολογίων. Η συγκεκριμένη μαθηματική μοντελοποίηση γενικοποιεί το μοντέλο των Bostel *et al.* (2008), και μπορεί να χρησιμοποιηθεί ως βάση για την ανάπτυξη παραλλαγών του ΠΔΟΠΠ.

### Το Προτεινόμενο Κυρίως Πρόβλημα

Έστω ότι  $\Omega_p$  είναι το σύνολο των εφικτών δρομολογίων για την περίοδο  $p$ . Οι συντελεστές  $a_{ir}^p$  ορίζονται ως εξής:

$$a_{ir}^p = \begin{cases} 1 & \text{εαν ο πελάτης } i \text{ περιλαμβάνεται στο δρομολόγιο } r \text{ της περιόδου } p \\ 0 & \text{αλλιώς} \end{cases} \quad (\text{Π.12})$$

Οι μεταβλητές  $x_r^p$  ορίζονται ως εξής:

$$x_r^p = \begin{cases} 1 & \text{αν το δρομολόγιο } r \text{ της περιόδου } p \text{ περιλαμβάνεται στη λύση} \\ 0 & \text{αλλιώς} \end{cases} \quad (\text{Π.13})$$

Εάν  $C_r^p$  είναι το κόστος του δρομολογίου  $r$  για την περίοδο  $p$ , η αντικειμενική συνάρτηση του ΚΠ είναι:

$$\min \sum_{p=1}^P \sum_{r \in \Omega_p} C_r^p x_r^p \quad (\text{Π.14})$$

υπό τους περιορισμούς:

$$\sum_{r \in \Omega_p} x_r^p \leq K \quad \forall p \in P \quad (\text{Π.15})$$

$$\sum_{p=1}^P \sum_{r \in \Omega_p} a_{ir}^p x_r^p \geq 1 \quad \forall i \in N \quad (\text{Π.16})$$

$$x_r^p = \{0, 1\} \quad (\text{Π.17})$$

Η αντικειμενική συνάρτηση (Π.14) αφορά στο συνολικό κόστος δρομολόγησης. Οι περιορισμοί (Π.15) αφορούν στο πλήθος των οχημάτων που μπορούν να χρησιμοποιηθούν σε κάθε περίοδο, ενώ οι περιορισμοί (Π.16) είναι οι περιορισμοί κάλυψης συνόλου (set covering). Τέλος, οι περιορισμοί (Π.17) δεσμεύουν τις μεταβλητές ροής σε δυαδικές

τιμές  $\{0, 1\}$  και η χαλάρωσή τους επιτρέπει την χρήση γνωστών μεθόδων γραμμικού προγραμματισμού.

Στο ανωτέρω μοντέλο έχουμε θεωρήσει ότι τα εφικτά δρομολόγια για κάθε περίοδο ( $\Omega_p$ ) είναι γνωστά *a priori*. Δεδομένου, όμως, ότι η υπόθεση αυτή δεν είναι πραγματοποιήσιμη στην πράξη λόγω του πλήθους των δυνατών συνδυασμών πελατών, ορίζουμε ως  $\Omega'_p$  ένα υποσύνολο του  $\Omega_p$ . Κάθε  $\Omega'_p$  περιέχει ένα περιορισμένο πλήθος εφικτών δρομολογίων της περιόδου  $p$ . Το ανωτέρω μοντέλο το οποίο αφορά το σύνολο  $\Omega'_p$ , αντί του  $\Omega_p$ , ορίζεται ως Περιορισμένο Κυρίως Πρόβλημα (ΠΚΠ) - Restricted Master Problem (RMP).

### Τα Υποπροβλήματα

Το κάθε ΥΠ αποτελεί ένα Στοιχειώδες Πρόβλημα Συντομότερης Διαδρομής με Χρονικά Παράθυρα και Περιορισμούς Χωρητικότητας (ΣΠΣΔΧΠΠΧ). Όλοι οι εναπομείναντες περιορισμοί μεταφέρονται στα ΥΠ ώστε να διασφαλίζεται η εφικτότητα των δρομολογίων. Η αντικειμενική συνάρτηση του ΣΠΣΔΧΠΠΧ αφορά στην εύρεση του δρομολογίου με το χαμηλότερο μειωμένο κόστος (reduced cost) για κάθε περίοδο  $p$ .

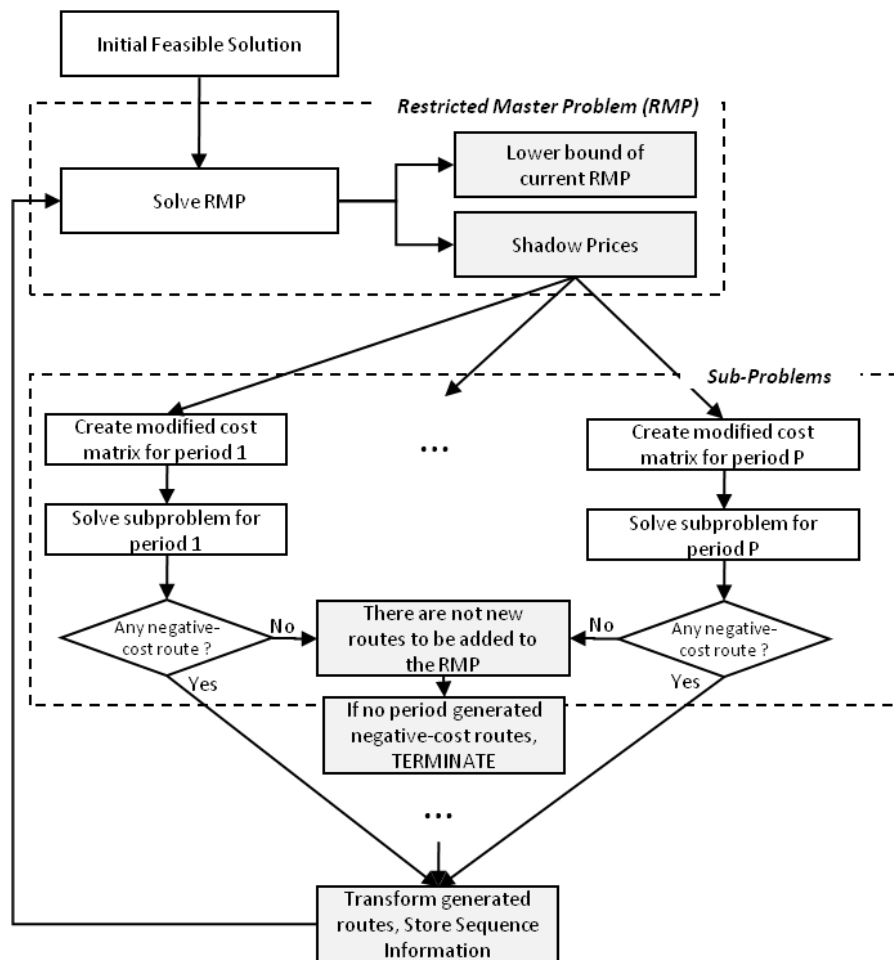
$$\min \sum_{i \in N_p \cup \{0\}} \sum_{j \in N_p \cup \{n+1\}} c'_{ij} x_{ijp} - \sigma_p \quad (\text{Π.18})$$

όπου το σύνολο  $N_p$  αποτελεί το σύνολο όλων των εφικτών πελατών εντός της περιόδου  $p$ , οι συντελεστές  $c'_{ij}$  ισούνται με  $c_{ij} - \pi_i$ , όπου οι συντελεστές  $\pi_i$  και  $\sigma_p$  είναι οι σκιώδεις τιμές (shadow prices), οι οποίες σχετίζονται με τους περιορισμούς (Π.14) και (Π.15) αντίστοιχα. Επισημαίνεται ότι το κάθε ΥΠ επιλύεται για κάθε μία περίοδο  $p$ . Καθότι το ΠΚΠ περιλαμβάνει μόνο εφικτά δρομολόγια, οι περιορισμοί (Π.4) έως (Π.10) εντάσσονται στο ΥΠ της κάθε περιόδου, λαμβάνοντας υπόψη τα σύνολα  $N_p$  αντί του συνόλου  $N$ .

### Συνδυάζοντας το Περιορισμένο Κυρίως Πρόβλημα με τα Υποπροβλήματα

Το Σχήμα Π.2 παρουσιάζει τη δομή της μεθόδου ΔΔΜ για τα ΠΔΟΠΠ. Επιλύοντας ένα ΠΚΠ, παρέχονται οι σχετικές σκιώδεις τιμές, σε συνδυασμό με την λύση του περιορισμένου προβλήματος (κόστος και σχετικά δρομολόγια). Οι σκιώδεις τιμές μεταφέρονται στα ΥΠ και χρησιμοποιούνται για τον υπολογισμό του κόστους  $c'_{ij}$ , για κάθε ακμή  $(i, j)$ . Στον πίνακα κόστους/αποστάσεων του ΣΠΣΔΧΠΠΧ χρησιμοποιούνται τα  $c'_{ij}$ , αντί των κανονικών τιμών κόστους. Οι σκιώδεις τιμές  $\sigma_p$  επίσης περιλαμβάνονται στον πίνακα κόστους του προβλήματος της κάθε περιόδου, μέσω της τροποποίησης  $c'_{0j} = c_{0j} - \sigma_p, \forall p \in P$ .

Επιλύοντας κάθε ΥΠ δημιουργείται ένα σύνολο δρομολογίων αρνητικού κόστους μείωσης. Κάθε ένα από αυτά τα δρομολόγια τροποποιείται σε μορφή κατάλληλη για το ΠΚΠ, στο οποίο χρησιμοποιούνται οι μεταβλητές  $a_{ir}^p$ . Επισημαίνεται ότι καθότι η αλληλουχία επίσκεψης των πελατών δεν διατηρείται στο ΠΚΠ, η πληροφορία αυτή πρέπει να διατηρείται χωριστά. Τα νέα δρομολόγια που προέκυψαν προστίθενται στα υφιστάμενα δρομολόγια εντός του ΠΚΠ το οποίο επιλύεται εκ νέου. Η διαδικασία αυτή τερματίζει όταν κανένα ΣΠΣΔΧΠΠΧ δεν μπορεί να δημιουργήσει επιπλέον δρομολόγια αρνητικού κόστους μείωσης. Στην περίπτωση αυτή, το ΠΚΠ επιστρέφει τη βέλτιστη λύση με το ελάχιστο κόστος δρομολόγησης και τα σχετικά δρομολόγια. Επισημαίνεται ότι η λύση αυτή είναι εν γένει μη ακέραια.



Σχήμα Π.2: Μέθοδος ΔΔΜ για ΠΔΟΠΠ

## ΑΠΟΔΟΤΙΚΕΣ ΤΕΧΝΙΚΕΣ ΓΙΑ ΤΟΝ ΥΠΟΛΟΓΙΣΜΟ ΤΟΥ ΚΑΤΩΤΑΤΟΥ ΟΡΙΟΥ (LOWER BOUND) ΤΟΥ ΠΔΟΠΠΧΠ

Με βάση την μέθοδο ΔΔΜ αναπτύχθηκαν δύο τεχνικές για την επιτάχυνση προσδιορισμού του κατώτατου ορίου του ΠΔΟΠΠΧΠ:

- Η τεχνική Cloning (CLONE), η οποία εκμεταλλεύεται την ευελιξία των πελατών να εξυπηρετηθούν (δρομολογηθούν) σε πολλαπλές περιόδους του ορίζοντα προγραμματισμού. Η μέθοδος αυτή μεταφέρει εφικτά δρομολόγια τα οποία δημιουργούνται από ένα ΥΠ σε άλλα ΥΠ της μεθόδου ΔΔΜ και στοχεύει στην μείωση του υπολογιστικού χρόνου αποφεύγοντας την επίλυση όλων των ΥΠ σε κάθε επανάληψη της μεθόδου ΔΔΜ. Παρόμοιες τεχνικές έχουν παρουσιαστεί από τους Pirkwieser κα Raidl (2009) και Mourgaya και Vanderbeck (2007) για το ΠΠΔΟ.
- Η τεχνική Unified (UNI) η οποία επιλύει ένα κοινό ΥΠ για όλες τις περιόδους του ορίζοντα προγραμματισμού. Η εφικτότητα του κάθε δρομολογίου εντός μίας περιόδου ελέγχεται εντός του κοινού ΥΠ. Η επίλυση του κοινού ΥΠ παρέχει όλα τα δρομολόγια για όλες τις περιόδους του ορίζοντα προγραμματισμού.

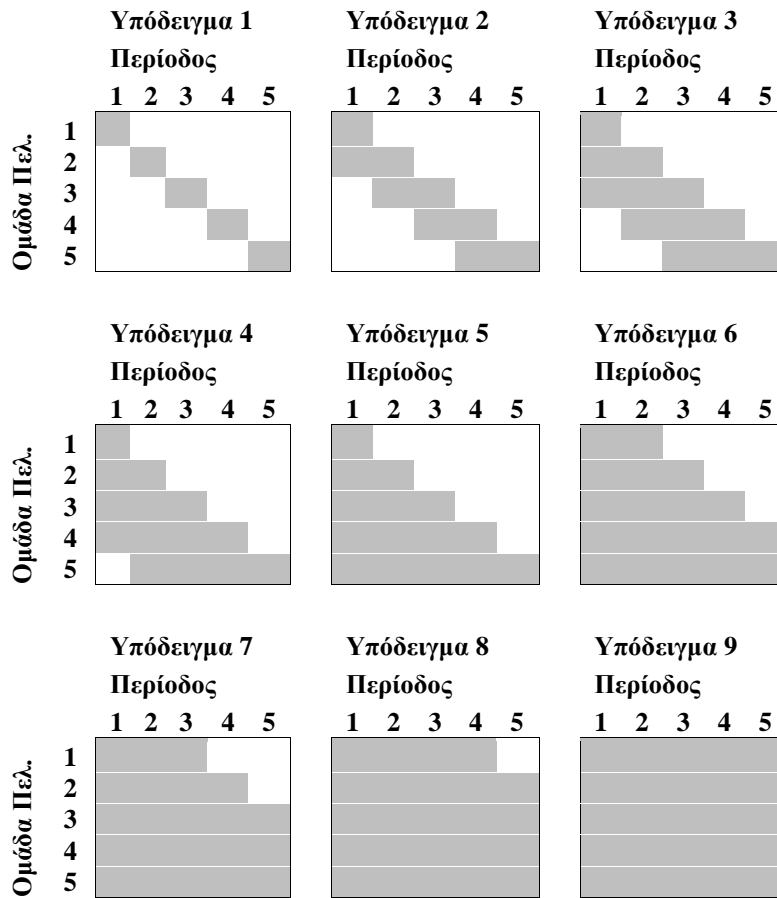
Η αποτελεσματικότητα των προτεινόμενων μεθόδων (σε σχέση με τον υπολογιστικό χρόνο) μελετήθηκε συγκριτικά α) με την κλασσική προσαρμογή (FULL) της ΔΔΜ για περιβάλλοντα δρομολόγησης πολλαπλών περιόδων (βλ. Σχήμα Π.2), καθώς και β) με υλοποίηση της κλασσικής μεθόδου σε περιβάλλον παράλληλης εφαρμογής (PARA).

### Πειραματική Διερεύνηση

Για την παραπάνω πειραματική διερεύνηση του ΠΔΟΠΠΧΠ δημιουργήθηκαν πειράματα με 50 πελάτες (παραγγελίες) βάσει των προβλημάτων R1, C1 και RC1 του Solomon. Για τη μετατροπή των προβλημάτων αυτών σε κατάλληλη μορφή για περιβάλλον πολλαπλών περιόδων, προστέθηκαν παράθυρα περιόδων, ως εξής:

- Ο ορίζοντας προγραμματισμού ορίστηκε σε πέντε (5) περιόδους
- Για κάθε πείραμα του Solomon, επιλεχτήκαν οι πρώτοι 50 πελάτες και διαχωρίστηκαν σε 5 ομάδες (10 πελάτες ανά ομάδα). Σε κάθε μία από τις ομάδες ανατέθηκε ένα διαφορετικό παράθυρο περιόδων
- Δημιουργήθηκαν εννέα υποδείγματα (μοτίβα) παραθύρων περιόδων, ώστε να μελετηθούν πειράματα με διαφορετική ευελιξία πελατών όσον αφορά τις εφικτές περιόδους δρομολόγησης. Έτσι, για κάθε ένα από τα πειράματα του Solomon, δημιουργήθηκαν εννέα διαφορετικά πειράματα.

Στο Σχήμα Π.3 παρουσιάζονται τα εννέα διαφορετικά υποδείγματα. Οι σκιασμένες περιοχές αφορούν στο παράθυρο περιόδων ανά υπόδειγμα και ομάδα πελατών.



Σχήμα Π.3: Υποδείγματα χρονικών παραθύρων περιόδων

Με βάση τα ανωτέρω υποδείγματα και τα πειράματα του Solomon, δημιουργήθηκαν 261 πειράματα σε περιβάλλον πολλαπλών περιόδων ως εξής:

- Για τη κατηγορία R1: 12 πειράματα x 9 υποδείγματα = 108 πειράματα ΠΔΟΠΠΧΠ
- Για τη κατηγορία C1: 9 πειράματα x 9 υποδείγματα = 81 πειράματα ΠΔΟΠΠΧΠ
- Για τη κατηγορία RC1: 8 πειράματα x 9 υποδείγματα = 72 πειράματα ΠΔΟΠΠΧΠ

Στον Πίνακα Π.2 παρουσιάζονται (α) ο αριθμός των πειραμάτων τα οποία επιλύθηκαν και για τα οποία βρέθηκε το κατώτατο όριο εντός συγκεκριμένου υπολογιστικού χρόνου μίας ώρας (με χρήση 8-πύρηνου υπολογιστή με επεξεργαστή 2GHz και 2GB μνήμης RAM), και (β) ο υπολογιστικός χρόνος ανά κατηγορία πειραμάτων και τεχνική επίλυσης.

Πίνακας Π.2: Υπολογιστικοί χρόνοι ανά κατηγορία πειραμάτων (ώρες)

Κατηγορία	Πειράματα	FULL	CLONE	UNI	PARA
R1	105	7,76	3,99	4,22	5,36
C1	73	1,47	0,77	1,29	0,85

Κατηγορία	Πειράματα	FULL	CLONE	UNI	PARA
RC1	71	5,90	3,34	3,04	3,64
Σύνολο	249	15,13	8,10	8,55	9,85

Οι εναλλακτικές τεχνικές (cloning and unified) επιτυγχάνουν μείωση του υπολογιστικού χρόνου σε σχέση με τις δύο άλλες μεθόδους, με εξαίρεση στα πειράματα με ομαδοποιημένους πελάτες (Clustered – C1). Συγκεκριμένα, για τις περιπτώσεις R1 και C1, η τεχνική CLONE επιτυγχάνει τα καλύτερα αποτελέσματα με μείωση του υπολογιστικού χρόνου κατά ~50% σε σχέση με τη μέθοδο FULL. Όσον αφορά στις περιπτώσεις RC1, η τεχνική UNI εμφανίζεται ως πιο αποτελεσματική, επιτυγχάνοντας, επίσης, μείωση του υπολογιστικού χρόνου κατά ~50%, ενώ συνολικά, η CLONE και η PARA εμφανίζονται ως οι πιο αποτελεσματικές μέθοδοι.

Στο Σχήμα Π.4 παρουσιάζονται οι μέσες τιμές των υπολογιστικών χρόνων ανά κατηγορία πειραμάτων (R1, C1 και RC1) και υπόδειγμα χρονικού παραθύρου περιόδων. Τα σχετικά αποτελέσματα συνοψίζονται στον Πίνακα Π.3.

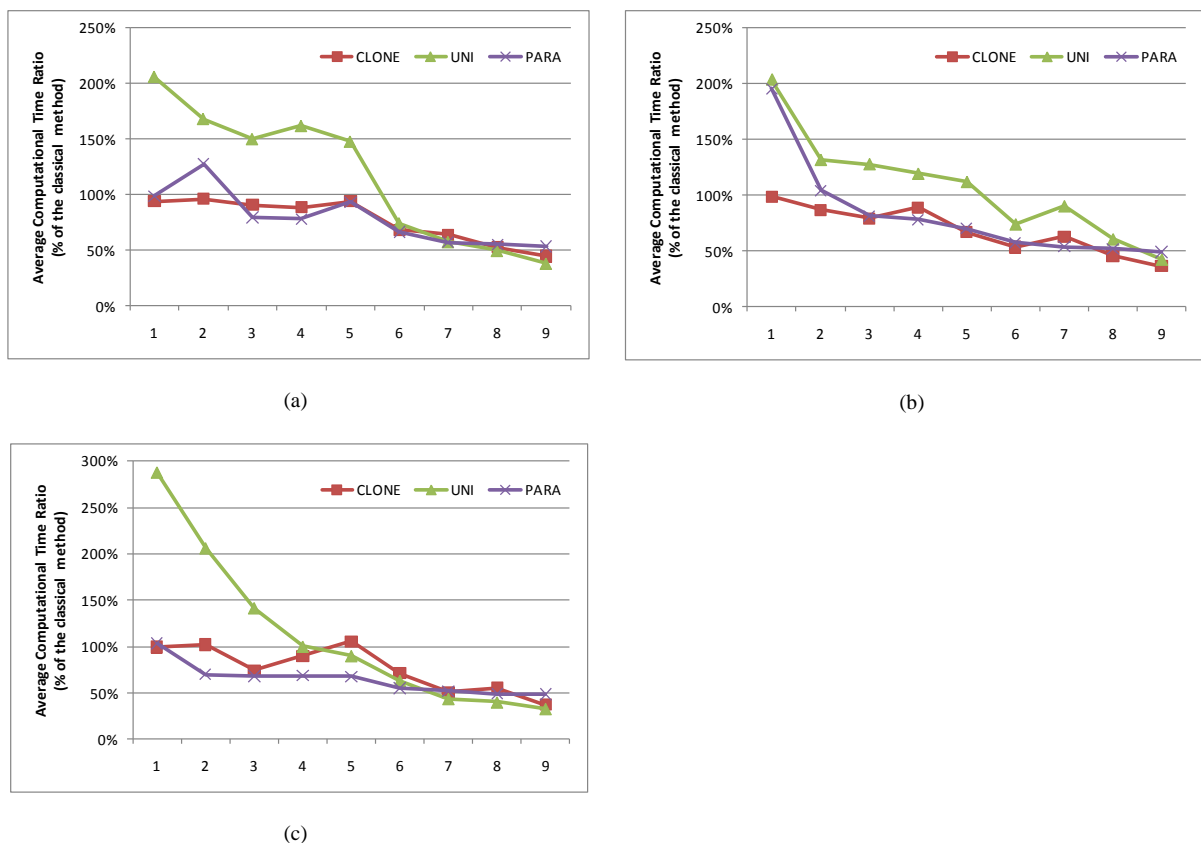


Figure Π.4: Μέσες τιμές υπολογιστικών χρόνων ανά υπόδειγμα χρονικού παραθύρου περιόδων και κατηγορία πειραμάτων (α) R1, (β) C1 και (γ) RC1.

Για κάθε ένα από τα υποδείγματα χρονικών παραθύρων περιόδων, ο Πίνακας Π.3 παρουσιάζει τον μέσο υπολογιστικό χρόνο της μεθόδου FULL, καθώς και το μέσο λόγο (%) του υπολογιστικού χρόνου των άλλων μεθόδων σε σχέση με τη μέθοδο FULL.

Πίνακας Π.3: Σύγκριση υπολογιστικού χρόνου

Υπόδειγμα	Μέσος Χρόνος FULL (σε δευτ.)	Μέσος Λόγος vs. FULL (%)		
		CLONE	UNI	PARA
1	4.2	97%	228%	130%
2	23.1	95%	169%	104%
3	53.4	83%	141%	77%
4	86.6	89%	132%	76%
5	108.8	89%	119%	78%
6	283.5	65%	71%	61%
7	434.5	60%	63%	55%
8	438.7	51%	50%	53%
9	525.9	40%	38%	51%

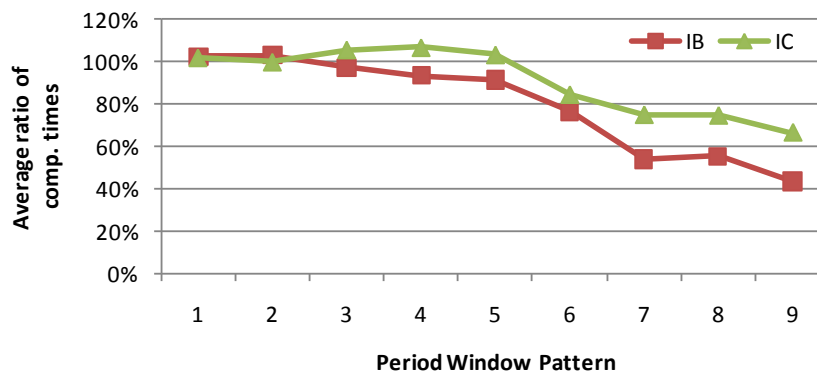
Όπως αναμενόταν, καμία μέθοδος δεν επιτυγχάνει σημαντική μείωση του υπολογιστικού χρόνου για τις περιπτώσεις των ιδιαίτερα στενών παραθύρων περιόδων. Ωστόσο, σημαντική μείωση παρατηρείται στις περιπτώσεις ευρέων παραθύρων περιόδων (υποδείγματα 6 έως 9), οι οποίες είναι και οι περιπτώσεις που απαιτούν τους μεγαλύτερους χρόνους υπολογισμού. Η UNI παρουσιάζει την πλέον ποικίλη συμπεριφορά σε σχέση με τα παράθυρα εφικτών περιόδων, ήτοι παρουσιάζει τη χαμηλότερη αποτελεσματικότητα στις περιπτώσεις στενών παραθύρων περιόδων (με χρόνους υπολογισμού μέχρι και 2 φορές μεγαλύτερων αυτών της μεθόδου FULL για την περίπτωση του υποδείματος 1). Ωστόσο, για τις περιπτώσεις των ευρύτερων παραθύρων περιόδων, υπερέχει των υπολοίπων μεθόδων επιτυγχάνοντας μείωση 62% σε σχέση με τη μέθοδο FULL για το υπόδειγμα 9.

## ΕΥΡΕΣΗ ΑΚΕΡΑΙΩΝ ΛΥΣΕΩΝ

Όπως επισημάνθηκε και ανωτέρω, ακέραιες λύσεις στο ΠΔΟΠΠΧΠ παρέχονται μέσω μεθόδου branch-and-price (B&P) η οποία είναι κατάλληλη για το περιβάλλον πολλαπλών περιόδων. Αναπτύχθηκαν και μελετήθηκαν δύο στρατηγικές για την διερεύνηση του δένδρου B&P: (α) η κλασσική στρατηγική ( $2br$ ), κατά την οποία για κάθε μη ακέραια λύση δημιουργούνται δύο διαφορετικά «κλαδιά» και (β) παραλλαγή η οποία θεωρεί  $P + 1$  κλαδιά λαμβάνοντας υπόψη τα χαρακτηριστικά πολλαπλών περιόδων του προβλήματος.

Για την μελέτη των ακέραιων λύσεων, συγκρίνουμε την μέθοδο B&P με τη τεχνική CLONE σε σχέση με την B&P με την τεχνική FULL. Η επιλογή της τεχνικής CLONE έναντι της UNI

βασίστηκε στο ότι η πρώτη (i) παρουσιάζει τους καλύτερους συνολικούς υπολογιστικούς χρόνους και (ii) είναι συνεπέστερη σε σχέση με όλα τα εναλλακτικά υποδείγματα παραθύρων περιόδων και όλες τις διαφορετικές γεωγραφικές κατανομές πελατών. Στο Σχήμα Π.5 παρουσιάζονται οι μέσες τιμές των λόγων των υπολογιστικών χρόνων της μεθόδου B&P, χρησιμοποιώντας την τεχνική CLONE σε σχέση με την FULL για όλα τα εναλλακτικά υποδείγματα παραθύρων περιόδων και όλες τις διαφορετικές γεωγραφικές κατανομές πελατών. Το Σχήμα παρουσιάζει δύο διαφορετικούς λόγους: (α) Ο λόγος IB αντιστοιχεί στα 196 πειράματα για τα οποία και οι δύο μέθοδοι εντόπισαν ακέραια λύση (είτε βέλτιστη είτε υπο-βέλτιστη). (β) Ο λόγος IC ο οποίος λαμβάνει υπόψη του μόνο τα πειράματα για τα οποία βρέθηκε η βέλτιστη λύση και από τις δύο μεθόδους.



Σχήμα Π.5: Μέσος λόγος υπολογιστικού χρόνου (CLONE vs. FULL)

Με βάση το ανωτέρω Σχήμα, η μέθοδος CLONE επιτυγχάνει μείωση του υπολογιστικού χρόνου για τον προσδιορισμό ακέραιων λύσεων, όσο διευρύνονται τα χρονικά παράθυρα περιόδων. Ωστόσο, η μείωση αυτή μετριάζεται για τις περιπτώσεις εκείνες στις οποίες προσδιορίστηκε η βέλτιστη λύση εντός του προκαθορισμένου χρονικού ορίου υπολογισμού. Το γεγονός αυτό μπορεί να αποδοθεί στο ότι το υπολογιστικό κέρδος το οποίο επιτυγχάνεται με την μέθοδο CLONE για την εύρεση του κατώτατου ορίου (lower bound) μετριάζεται από τον υπολογισμό των λοιπών B&P κλάδων, στους οποίους δημιουργείται περιορισμένος αριθμός κολωνών (δρομολόγια) και, συνεπώς, η μέθοδος CLONE δε δύναται να αποφέρει υπολογιστικό κέρδος.

### Ευρετική Τεχνική Απόρριψης Κλάδων του Δένδρου B&P

Για την εύρεση «αποδοτικών» λύσεων σε συντομότερο χρονικό διάστημα, προτείνουμε μέθοδο κατά την οποία παύει η περαιτέρω επίλυση εκείνων των κόμβων του δένδρου B&P, για τους οποίους το κατώτατο όριο (lower bound) έχει μικρή απόκλιση από το καλύτερο

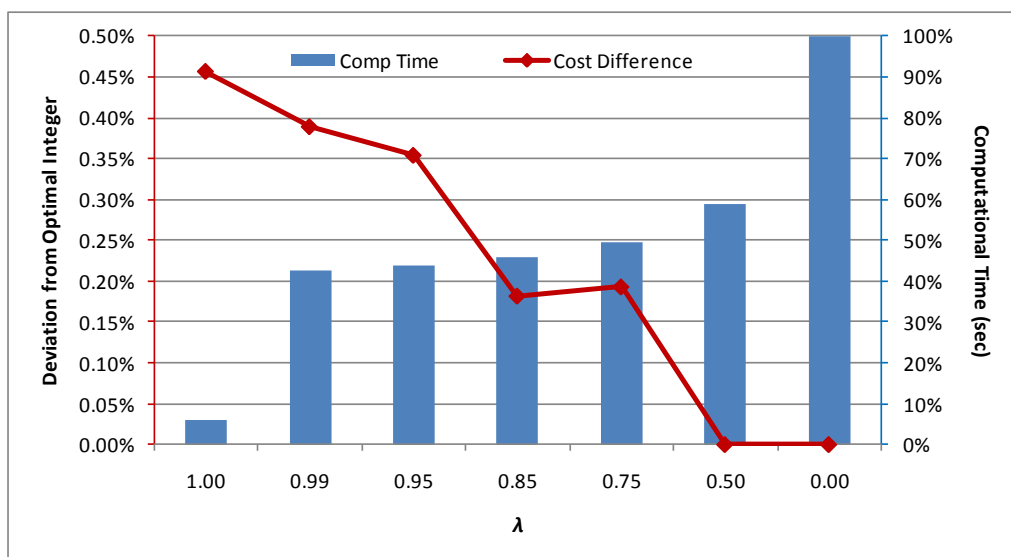
γενικό ακέραιο ανώτατο όριο (Global Upper Bound) που έχει βρεθεί έως εκείνη τη στιγμή. Με τον τρόπο αυτό επιτυγχάνεται η εύρεση ποιοτικών υπο-βέλτιστων λύσεων, ακόμα και σε περιπτώσεις με ευρέα παράθυρα περιόδων.

Η διαδικασία της προτεινόμενης μεθόδου έχει ως εξής: Με την επίλυση ενός κόμβου (έστω του κόμβου  $n$ ) του δένδρου  $B\&P$  βρίσκεται το κατώτατο όριο  $LB_n$ . Επιπρόσθετα, ένα ανώτατο ακέραιο όριο του κόμβου  $n$  (έστω  $IB_n$ ) υπολογίζεται επιλύοντας το πρόβλημα B&B μόνο με τα δρομολόγια τα οποία περιέχονται ήδη στο τρέχον ΠΚΠ (μέσω του υπολογιστικού εργαλείου CPLEX). Δεδομένης της καλύτερης γνωστής ακέραιης λύσης (έστω  $IUB$ ) που είναι γνωστή έως εκείνη τη στιγμή, υπολογίζεται για κάθε κόμβο  $n$  η παρακάτω μετρική :

$$M_n = \frac{IUB - LB_n}{IB_n - LB_n} \quad (\Pi.18)$$

Διατηρούνται και επιλύονται περαιτέρω μόνο οι κόμβοι εκείνοι για τους οποίους ισχύει ότι  $M_n > \lambda$  για  $\lambda \in [0,1]$ .

Για την μελέτη της αποτελεσματικότητας μεθόδου, εστιάζουμε στις 66 περιπτώσεις για τις οποίες η B&P σύγκλινε στην βέλτιστη λύση εντός του προκαθορισμένου χρονικού ορίου υπολογισμού. Το Σχήμα Π.6 παρουσιάζει τα αποτελέσματα σε σχέση τόσο με τον υπολογιστικό χρόνο όσο και με την απόκλιση από την βέλτιστη λύση για διαφορετικές τιμές της παραμέτρου  $\lambda$  χρησιμοποιώντας την μέθοδο  $P + 1$ . Ο υπολογιστικός χρόνος είναι κανονικοποιημένος σε σχέση με τον χρόνο της μεθόδου B&P με  $\lambda = 0$ . Επισημαίνεται ότι για την απόκλιση από τη βέλτιστη λύση χρησιμοποιήθηκαν μόνο οι περιπτώσεις οι οποίες δεν συνέκλιναν σε αυτή.



Σχήμα Π.6: Υπολογιστικοί χρόνοι και απόκλιση από την βέλτιστη ακέραια λύση για διαφορετικές τιμές της παραμέτρου  $\lambda$  (μέθοδος  $P + 1$ )

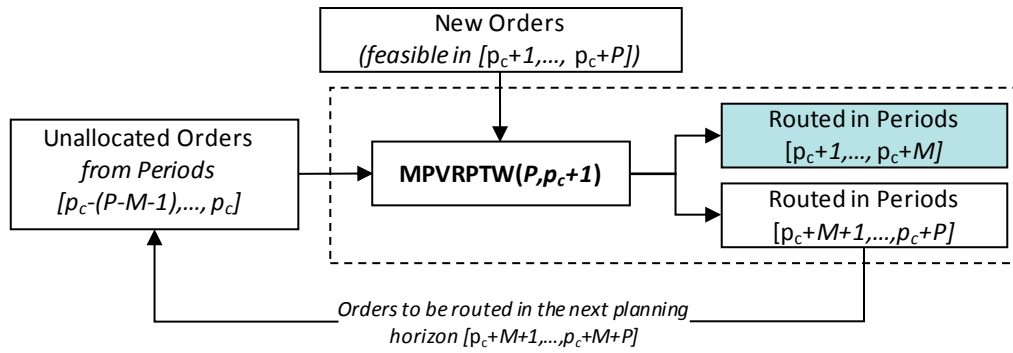
Τα αποτελέσματα του Σχ. Π.6 επιβεβαιώνουν την αποδοτικότητα της μεθόδου καθώς η βέλτιστη ακέραια λύση επιτυγχάνεται για  $\lambda > 0.5$  στο 60% του χρόνου τον οποίο απαιτεί η ολοκληρωμένη μέθοδος B&P. Επιπρόσθετα, η απόκλιση από την βέλτιστη λύση είναι περιορισμένη και ελεγχόμενη από την τιμή της παραμέτρου  $\lambda$ . Επισημαίνεται ότι ακόμα και για  $\lambda = 1$ , η απόκλιση του κόστους είναι μικρότερη του 0.5%, ενώ η μείωση του χρόνου ανέρχεται σε 94%.

## ΠΕΡΙΒΑΛΛΟΝ ΚΥΛΙΟΜΕΝΟΥ ΧΡΟΝΙΚΟΥ ΟΡΙΖΟΝΤΑ

Στη συνέχεια μελετήθηκε το ΠΔΟΠΠΧΠ σε εκτεταμένο χρονικό ορίζοντα και η επίλυσή του μέσω προγραμματισμού κυλιόμενου ορίζοντα. Έστω ότι ένας πελάτης (παραγγελία)  $i$  γίνεται γνωστός την περίοδο  $t$  και μπορεί να εξυπηρετηθεί εντός του παραθύρου  $[\xi_i^s, \xi_i^e]$ , όπου  $\xi_i^s > t$ . Ο συνολικός μακροχρόνιος ορίζοντας του προβλήματος (ΜΧΟ) καθορίζεται από την μέγιστη τελική περίοδο,  $\xi_i^e$ , έως την οποία μπορεί να εξυπηρετηθεί οποιοσδήποτε πελάτης  $i \in N$ , και ορίζεται ως  $S$ . Επισημαίνεται ότι ο ορίζοντας  $S$  εξαρτάται από το σύνολο πελατών  $N$ , αλλά για λόγους απλούστευσης δεν συμπεριλαμβάνεται το  $N$  στο σύμβολο  $S$ .

Έστω  $\Pi\Delta\Omega\text{ΠΠΧΠ}(P, p_c + 1)$  το σχετικό πρόβλημα εντός ορίζοντα  $[p_c + 1, p_c + P]$  μήκους  $P < S$  περιόδων. Οι πελάτες οι οποίοι περιλαμβάνονται στο πρόβλημα αυτό (σύνολο πελατών  $\bar{N}$ ) είναι αυτοί για τους οποίους το παράθυρο εφικτών περιόδων αρχίζει εντός του ορίζοντα προγραμματισμού, δηλ.,  $\bar{N} = \{i \in N: p_c + 1 \leq \xi_i^s \leq p_c + P\}$ , και  $(\bar{N} \subseteq N)$ .

Η μέθοδος προσέγγισης του προβλήματος έχει ως εξής: Οι πελάτες ανατίθενται στις επόμενες  $P$  περιόδους, δηλ. στις περιόδους  $[p_c + 1, p_c + P]$  επιλύοντας το  $\Pi\Delta\Omega\text{ΠΠΧΠ}(P, p_c + 1)$ . Το μήκος ( $P$ ) του ορίζοντα προγραμματισμού επιλέγεται ώστε να παρέχει ικανοποιητικές λύσεις «βλέποντας» στο μέλλον, αλλά και ώστε να μην καθιστά απαγορευτικό τον απαιτούμενο για την επίλυση του προβλήματος υπολογιστικό χρόνο. Με βάση τη λύση του  $\Pi\Delta\Omega\text{ΠΠΧΠ}(P, p_c + 1)$ , επιλέγονται προς εξυπηρέτηση οι πελάτες οι οποίοι ανατέθηκαν στις περιόδους  $[p_c + 1, \dots, p_c + M]$ , όπου  $M \leq P$ . Οι εναπομείναντες πελάτες (οι οποίοι ανατέθηκαν στο διάστημα  $[p_c + M + 1, p_c + M + P]$ ) δρομολογούνται ξανά σε συνδυασμό με τους νέους πελάτες, το παράθυρο περιόδων των οποίων αρχίζει εντός των περιόδων  $p_c + 1, \dots, p_c + M$ . Η κυλιόμενη αυτή διαδικασία προγραμματισμού παρουσιάζεται στο Σχήμα Π.7.



Σχήμα Π.7. Διαδικασία προγραμματισμού

Με βάση το αναφερόμενο περιβάλλον κυλιόμενου χρονικού ορίζοντα, μελετώνται δύο περιπτώσεις:

Η **ημι-στατική** περίπτωση, στην οποία όλοι οι πελάτες εντός του ορίζοντα  $S$  θεωρούνται γνωστοί. Χρησιμοποιώντας ορίζοντα προγραμματισμού μήκους  $P$  περιόδων και ορίζοντα υλοποίησης μήκους  $M$  περιόδων, ο κύκλος επίλυσης επαναλαμβάνεται κάθε  $M$  περιόδους. Στην περίπτωση αυτή, οι επιπρόσθετοι πελάτες για κάθε επόμενο ΠΔΟΠΠΧΠ είναι οι πελάτες των τελευταίων  $M$  περιόδων του αντίστοιχου (νέου) ορίζοντα προγραμματισμού.

Η **δυναμική** περίπτωση, στην οποία σε κάθε περίοδο εμφανίζονται νέοι πελάτες. Στην περίπτωση αυτή δεν είναι γνωστοί όλοι οι πελάτες εντός του ορίζοντα προγραμματισμού. Πλήρης γνώση των πελατών υπάρχει μόνο για την πρώτη περίοδο του ορίζοντα προγραμματισμού.

### Θεωρητική Διερεύνηση

Για την ημι-στατική περίπτωση, διερευνήθηκε η επίδραση δύο βασικών παραμέτρων του προγραμματισμού κυλιόμενου χρονικού ορίζοντα: Ο ορίζοντας υλοποίησης ( $M$ ) και ο ορίζοντας προγραμματισμού ( $P$ ).

Έστω ότι  $ΠΠ(P, p_c + 1)$  είναι η βέλτιστη λύση του  $ΠΔΟΠΠΧΠ(P, p_c + 1)$ , και  $C(P, p_c + 1)$  είναι το σχετικό βέλτιστο κόστος, το οποίο αφορά στο συνολικό αθροιστικό κόστος δρομολόγησης εντός των περιόδων του ορίζοντα προγραμματισμού:  $C(P, p_c + 1) = \sum_{\omega=p_c+1}^{\min(p_c+P, S)} C(P, \omega)$ , όπου  $C(P, \omega)$  είναι το κόστος δρομολόγησης της περιόδου  $\omega$ . Ορίζουμε ως  $\bar{C}_{PM}^S$  το τελικό υλοποιηθέν κόστος δρομολόγησης του συνολικού μακροχρόνιου ορίζοντα  $S$ , το οποίο προκύπτει μέσω τεχνικής κυλιόμενου χρονικού ορίζοντα προγραμματισμού μήκους  $P$  και ορίζοντα υλοποίησης μήκους  $M$  περιόδων, αντίστοιχα. Για λόγους απλοποίησης θεωρούμε ότι το  $P$  είναι ακέραιο πολλαπλάσιο του  $M$ . Τότε

$$\bar{C}_{PM}^S = \sum_{p=1}^{\frac{S}{M}} \sum_{k=1}^M C(P, p * M + k) \quad (\text{Π.19})$$

Έστω, επίσης, σύνολο εφικτών πελατών εντός της περιόδου  $p$  και θεωρούμε το βέλτιστο κόστος δρομολόγησης για την εξυπηρέτηση των πελατών αυτών. Στην περίπτωση που υπάρχουν πολλαπλές λύσεις με το ίδιο βέλτιστο κόστος, επιλέγεται μία τυχαία. Ορίζουμε ως  $O_p$  το σύνολο των βέλτιστων λύσεων (συνδυασμό δρομολογίων) για διαφορετικά υποσύνολα πελατών, εφικτών εντός της περιόδου  $p$ . Επισημαίνεται ότι για κάθε διαφορετικό υποσύνολο πελατών μόνο μία λύση εμπεριέχεται εντός του  $O_p$ .

Για την ημιστατική περίπτωση και με βάση τους παραπάνω ορισμούς έχουμε διατυπώσει και αποδείξει τις παρακάτω *διαπιστώσεις*.

Η 1<sup>η</sup> *Διαπίστωση* αποδεικνύει την αποτελεσματικότητα της μονολιθικής επίλυσης του συνολικού προβλήματος δρομολόγησης (για τον μακροχρόνιο ορίζοντα  $S$ ) σε σχέση με κάθε άλλη πιθανή λύση η οποία λαμβάνεται μέσω της τεχνικής κυλιόμενου χρονικού ορίζοντα.

Η 2<sup>η</sup> *Διαπίστωση* σχετίζεται με το μήκος του ορίζοντα προγραμματισμού  $P$  που χρησιμοποιείται στην τεχνική κυλιόμενου χρονικού ορίζοντα. Αποδεικνύεται ότι ευρύτεροι ορίζοντες προγραμματισμού δεν είναι απαραίτητο ότι θα οδηγήσουν και σε καλύτερες λύσεις.

Τέλος, η 3<sup>η</sup> *Διαπίστωση* δείχνει ότι ένας συντομότερος ορίζοντας υλοποίησης  $M$  δεν οδηγεί αναγκαστικά σε καλύτερη τελική λύση δρομολόγησης.

### **Τροποποιήσεις του ΠΔΟΠΠΧΠ για την εφαρμογή του σε περιβάλλον Κυλιόμενου Χρονικού Ορίζοντα**

Για την εφαρμογή του ΠΔΟΠΠΧΠ σε περιβάλλον κυλιόμενου χρονικού ορίζοντα απαιτούνται συγκεκριμένες τροποποιήσεις τόσο στην αντικειμενική συνάρτηση, όσο και στη μέθοδο επίλυσης. Οι τροποποιήσεις αυτές αφορούν στη δυνατότητα να αναβάλλεται η εξυπηρέτηση των πελατών από τον ένα ορίζοντα προγραμματισμού στον επόμενο. Για τον λόγο αυτό εισάγουμε κατάλληλες συναρτήσεις ποινών (penalty functions).

#### Τροποποιήσεις της Αντικειμενικής Συνάρτησης

Έστω πρόβλημα ΠΔΟΠΠΧΠ με ορίζοντα προγραμματισμού  $[1, P]$ . Ορίζουμε ως  $u_e$  το σύνολο των μη δρομολογηθέντων εντός του ορίζοντα  $P$  πελατών, τα παράθυρα περιόδων των οποίων λήγουν την περίοδο 1, δηλ.  $\xi_i^S = 1$ , (υποχρεωτικοί πελάτες) και ως  $u_f$  το σύνολο των μη δρομολογηθέντων πελατών τα παράθυρα περιόδων των οποίων δεν λήγουν την περίοδο 1,

δηλ.  $\xi_i^s > 1$ , (μη υποχρεωτικοί πελάτες). Με βάση τα ανωτέρω προτείνεται η τροποποιημένη αντικειμενική συνάρτηση.

$$\min \left( \sum_{p=1}^P \sum_{r \in \Omega_p} C_r^p x_r^p \right) + P_e |u_e| + P_f |u_f| \quad (\text{Π.20})$$

όπου οι συντελεστές  $P_e$  και  $P_f$  είναι οι ποινές για κάθε μη δρομολογημένο πελάτη, υποχρεωτικό ή μη, αντίστοιχα. Εάν για τις συγκεκριμένες ποινές οριστούν τιμές χαμηλότερες από  $C_{r_i}$ , όπου  $C_{r_i}$  είναι το κόστος του μοναδιαίου δρομολογίου [ $Depot - i - Depot$ ], τότε υπάρχει πιθανότητα τα εικονικά δρομολόγια (κολώνες) εντός της ΔΔΜ τα οποία σχετίζονται με τις ποινές να εισέλθουν στη τελική βάση της λύσης σε βάρος των μοναδιαίων δρομολογίων και με αυτό τον τρόπο να μη δρομολογηθεί ο πελάτης  $i$ . (Επισημαίνεται ότι η ομαδοποίηση των πελατών σε κοινά δρομολόγια μπορεί να αποτρέψει τη συγκεκριμένη συμπεριφορά).

Ρυθμίζοντας τις Ποινές  $P_e$  και  $P_f$  ώστε να δίνεται Προτεραιότητα στους Υποχρεωτικούς Πελάτες

Με βάση τις προαναφερθείσες ποινές δεν γίνεται διάκριση των μη υποχρεωτικών πελατών ανάλογα με την περίοδο λήξης του καθενός (δηλ. με βάση την εγγύτητα της περιόδου λήξης). Με τον τρόπο αυτό οι μη υποχρεωτικοί πελάτες οι οποίοι θα δρομολογηθούν στην περίοδο 1 επιλέγονται αποκλειστικά με βάση το κόστος δρομολόγησης, και συνεπώς ενδέχεται υποχρεωτικοί πελάτες να μείνουν εκτός δρομολόγησης, οδηγώντας σε μειωμένο αριθμό εξυπηρετούμενων πελατών. Για να επιβεβαιώσουμε ότι ένας υποχρεωτικός πελάτης δεν θα αντικατασταθεί από έναν μη υποχρεωτικό, προτείνουμε την παρακάτω ανισότητα:

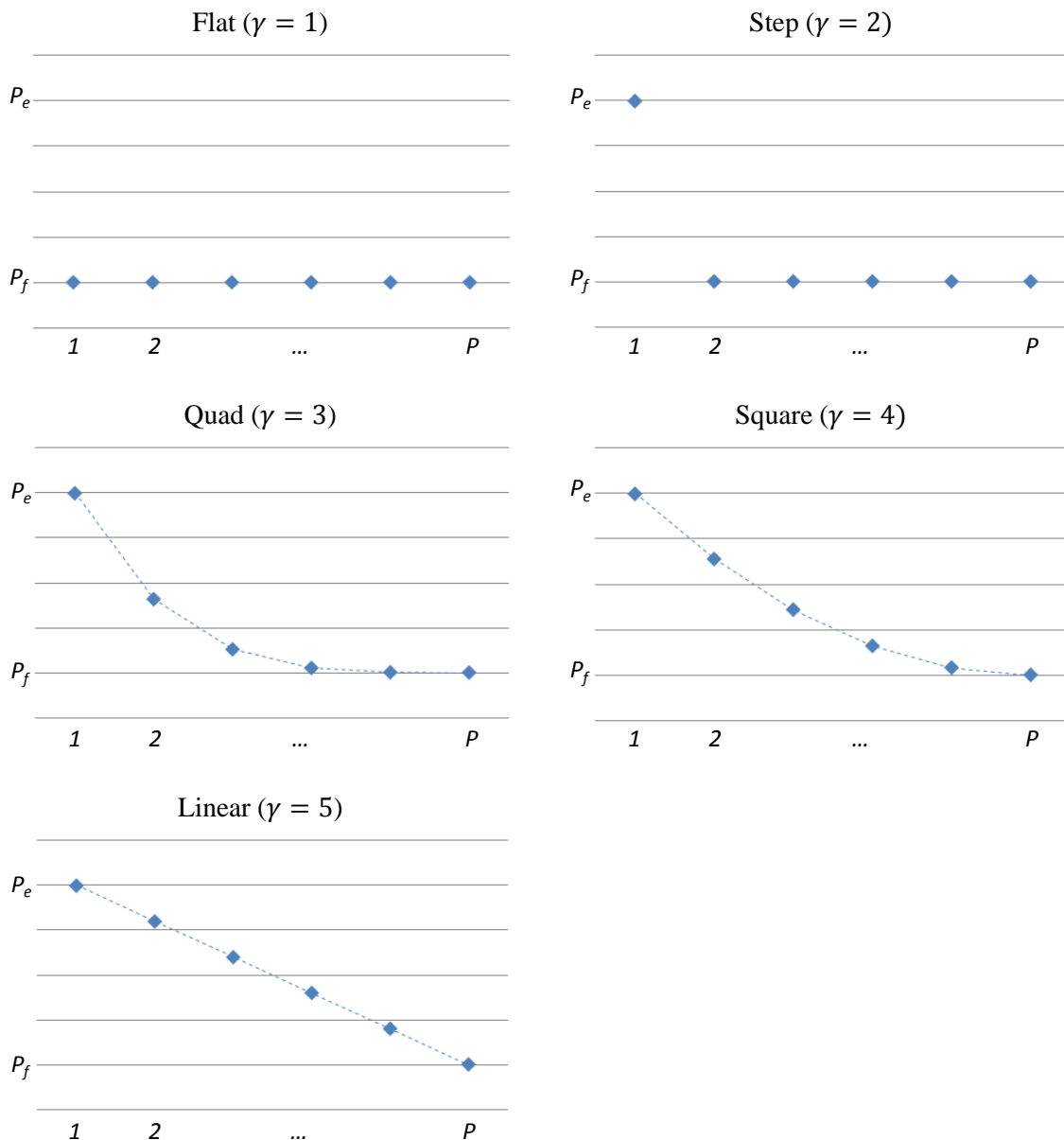
$$P_e \geq P_f + \Delta M_{e,f} \quad (\text{Π.21})$$

και ορίζουμε την κατάλληλη τιμή του  $\Delta M_{e,f}$  ώστε να εξυπηρετούνται όλοι οι υποχρεωτικοί πελάτες (με εξαίρεση όσους δε μπορούν να δρομολογηθούν λόγω περιορισμών πόρων:

$$\Delta M_{e,f} \geq \delta = \sum_{i=1}^N C_{r_i} + (\hat{f} - 1) \max_{i \in N} (C_{r_i}) + \vartheta \quad (\text{Π.22})$$

Όπου  $\hat{f}$  είναι ο αριθμός των (μη δρομολογηθέντων) μη υποχρεωτικών πελατών και  $\vartheta$  είναι ένας μικρός θετικός αριθμός. Αν το  $\delta$  οριστεί σε αυτήν την τιμή, τότε ο αριθμός των υποχρεωτικών πελατών που θα εξυπηρετηθούν θα είναι ο μέγιστος δυνατός.

Για να μελετηθούν διαφορετικές περιπτώσεις διάκρισης των μη υποχρεωτικών πελατών με βάση την εγγύτητα της λήξης του παραθύρου περιόδων, προτείνουμε πέντε (5) εναλλακτικές συναρτήσεις ποινής (penalty functions). Έτσι, η ποινή  $P_i^y$  που ανατίθεται σε κάθε πελάτη  $i$  εξαρτάται από την καταληκτική περίοδο ( $\xi_i^e$ ) και τη μορφή της συνάρτησης  $y$ . Στο Σχήμα Π.8 παρουσιάζονται οι πέντε συναρτήσεις ποινής. Χρησιμοποιώντας την κατάλληλη συνάρτηση, μπορούμε να κατευθύνουμε την μέθοδο επίλυσης ώστε να δίνει προτεραιότητα στους υποχρεωτικούς πελάτες, καθώς και στους πελάτες με περιορισμένη περιοδική ευελιξία (δηλ. περιορισμένο πλήθος διαθέσιμων περιόδων δρομολόγησης).

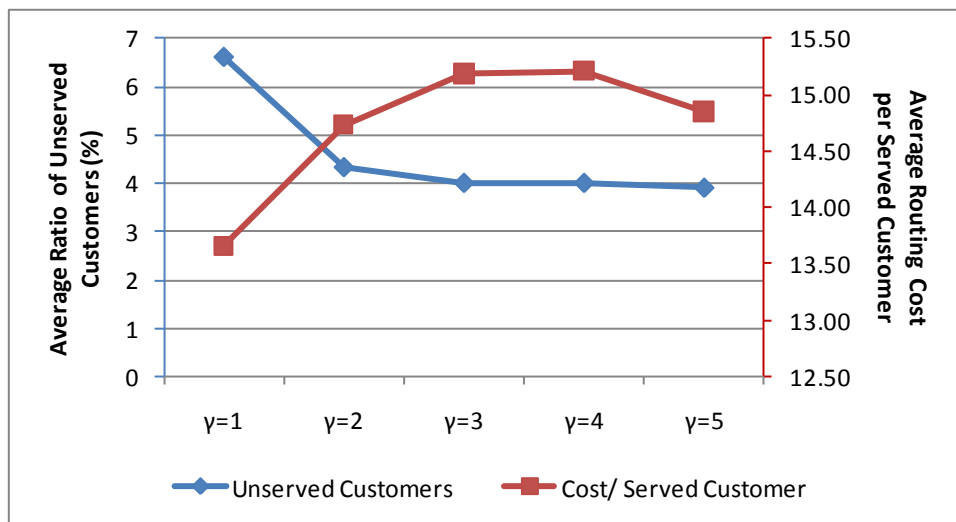


Σχήμα Π.8: Συναρτήσεις ποινής ( $\gamma = 1, \dots, 5$ )

Για την μελέτη της αποδοτικότητας των διαφορετικών συναρτήσεων χρησιμοποιήθηκαν οι παρακάτω μετρικές:

- Ποσοστό μη δρομολογημένων πελατών: Για κάθε συνδυασμό  $P$  και  $\gamma$  θεωρούμε το συνολικό πλήθος μη δρομολογημένων πελατών σε σχέση με το συνολικό πλήθος πελατών.
- Κόστος δρομολόγησης ανά δρομολογημένο πελάτη (λόγος κόστους δρομολόγησης): Αφορά το συνολικό κόστος δρομολόγησης διαιρεμένο με τον αριθμό (πλήθος) των δρομολογημένων πελατών.

Στο Σχήμα Π.9 παρουσιάζεται ο μέσος λόγος των μη δρομολογημένων πελατών και ο λόγος του κόστους δρομολόγησης για οριζόντες προγραμματισμού  $P = 1$  έως  $5$ , και για κάθε διαφορετική συνάρτηση ποινής ( $\gamma = 1$  ; έως  $5$ ).



Σχήμα Π.9: Μη δρομολογημένοι πελάτες και κόστος δρομολόγησης (μέσες τιμές) για τις 5 διαφορετικές συναρτήσεις ποινής

Όπως φαίνεται από το Σχήμα Π.9, οι συναρτήσεις  $\gamma = 3, 4$  και  $5$  επιτυγχάνουν αύξηση των δρομολογημένων πελατών καθώς δίνουν προτεραιότητα στους υποχρεωτικούς πελάτες αλλά και στους πελάτες με περιορισμένη ευελιξία περιόδων. Για τις συναρτήσεις αυτές είναι λογικό να αναμένουμε αύξηση του κόστους δρομολόγησης. Επισημαίνεται ότι η συνάρτηση  $\gamma = 5$  επιτυγχάνει χαμηλότερο κόστος δρομολόγησης ανάμεσα στις τρεις αυτές συναρτήσεις καθώς επιτρέπει περισσότερη ευελιξία κατά την βελτιστοποίηση του κόστους δρομολόγησης. Για τον λόγο αυτό χρησιμοποιείται στη συνέχεια της παρούσας έρευνας.

### Πειραματική Διερεύνηση Κυλιόμενου Χρονικού Ορίζοντα

Σκοπός της ανάλυσης είναι να ελεγχθεί η επίδραση του ορίζοντα προγραμματισμού  $P$  και του ορίζοντα υλοποίησης  $M$  σε σχέση με δύο μετρικές απόδοσης: (α) Τον αριθμό των δρομολογημένων πελατών, και (β) το λόγο κόστους δρομολόγησης. Στην πειραματική διερεύνηση χρησιμοποιήθηκαν πειράματα με διαφορετικές γεωγραφικές κατανομές (R1, C1, RC1) και διαφορετικό εύρος χρονικών παραθύρων. Τα χαρακτηριστικά των περιπτώσεων που αναλύθηκαν και στις δύο περιπτώσεις, ημι-στατική και δυναμική, έχουν ως εξής:

- Σε κάθε πείραμα χρησιμοποιείται ορίζοντας 30 περιόδων και 300 πελάτες
- Για τα παράθυρα περιόδων χρησιμοποιείται το υπόδειγμα 3 καθότι παρέχει μετριασμένη περιοδική ευελιξία πελατών
- Δύο (2) οχήματα θεωρήθηκαν ως διαθέσιμα για κάθε περίοδο
- Για κάθε πείραμα μελετήθηκαν δύο διαφορετικοί ορίζοντες υλοποίησης ( $M = 1$  και  $M = 2$ ) και δύο ορίζοντες προγραμματισμού ( $N = 3$  και  $N = 5$ )
- Το ΠΔΟΠΠΧΠ επιλύεται χρησιμοποιώντας τις παρακάτω παραμέτρους:
  - Το κατώτατο όριο υπολογίζεται με την μέθοδο Cloning
  - Χρησιμοποιήθηκε η γραμμική συνάρτηση ποινής (δηλ.  $\gamma = 5$ )
  - Οι ακέραιες λύσεις υπολογίστηκαν με την ευρετική μέθοδο με  $\lambda = 1$ .

### Πειραματικά Αποτελέσματα για την Ημι-στατική Περίπτωση

Ο Πίνακας Π.4 παρουσιάζει για κάθε πείραμα, το πλήθος των δρομολογημένων πελατών, καθώς και το μέσο κόστος δρομολόγησης ανά πελάτη για το συνολικό ορίζοντα των 30 περιόδων. Οι τιμές αυτές δίνονται για  $P = 3$  και 5, και  $M = 1$  και 2.

Πίνακας Π.4: Συγκριτικά αποτελέσματα για την επίλυση της ημι-στατικής περίπτωσης με διαφορετικούς ορίζοντες προγραμματισμού και υλοποίησης

Πείραμα	P	Δρομολογημένοι Πελάτες		Κόστος Δρομολόγησης/Πελάτη	
		M=1	M=2	M=1	M=2
L_r103	3	293	287	20.09	20.62
	5	293	292	19.26	19.43
L_r106	3	299	298	19.54	20.03
	5*	299	289	18.31	18.17
L_r109	3	299	294	19.77	20.13
	5	299	293	17.62	18.27
L_c106	3	300	298	27.90	28.88
	5*	300	296	25.78	25.32
L_c108	3	300	300	22.72	24.41
	5*	300	297	21.68	21.59

Πείραμα	P	Δρομολογημένοι Πελάτες		Κόστος Δρομολόγησης/Πελάτη	
		M=1	M=2	M=1	M=2
L_c102	3	299	296	26.64	27.22
	5	299	298	24.40	24.52
L_rc101	3*	283	240	28.86	31.98
	5*	283	241	28.36	28.88
L_rc105	3*	293	253	26.84	27.95
	5*	294	267	25.52	27.28
L_rc107	3	300	298	23.35	23.76
	5	300	300	21.00	21.16
Μέσος Όρος	3	296.2	284.9	23.97	25.00
	5	296.3	285.9	22.44	22.74
Χρονικά	Στενά	292.0	275.7	25.04	25.85
Παράθυρα	Μέτρια	297.5	284.0	22.43	23.24
	Μεγάλα	299.3	296.5	22.13	22.51

\* Περιπτώσεις στις οποίες ο ορίζοντας υλοποίησης  $M = 2$  επιτυγχάνει χαμηλότερο κόστος δρομολόγησης σε σχέση με τον ορίζοντα υλοποίησης  $M = 1$ .

Ο ευρύτερος ορίζοντας προγραμματισμού επιτυγχάνει μείωση του κόστους δρομολόγησης, επιβεβαιώνοντας την καταλληλότητα των προτεινόμενων μεθόδων. Όσον αφορά στον ορίζοντα υλοποίησης  $M$ , είναι σαφές ότι η περίπτωση  $M = 1$  επιτυγχάνει υψηλότερο (ή ίσο) αριθμό δρομολογημένων πελατών σε σχεδόν όλες τις περιπτώσεις, με εξαίρεση τις περιπτώσεις στις οποίες η  $M = 2$  εξυπηρετεί πολύ λιγότερους πελάτες. Τα αποτελέσματα αυτά παραμένουν συνεπή για διαφορετικά χρονικά παράθυρα, καθώς και για διαφορετικές γεωγραφικές κατανομές πελατών.

#### Πειραματικά Αποτελέσματα για την Δυναμική Περίπτωση

Για τα συγκεκριμένα πειράματα χρησιμοποιήθηκε μόνο η περίπτωση  $M=1$ . Στον Πίνακα Π.5 παρουσιάζονται ο αριθμός των δρομολογημένων πελατών και το μέσο κόστος δρομολόγησης ανά πελάτη για τον ορίζοντα των 30 περιόδων και για τις δύο τιμές του ορίζοντα προγραμματισμού.

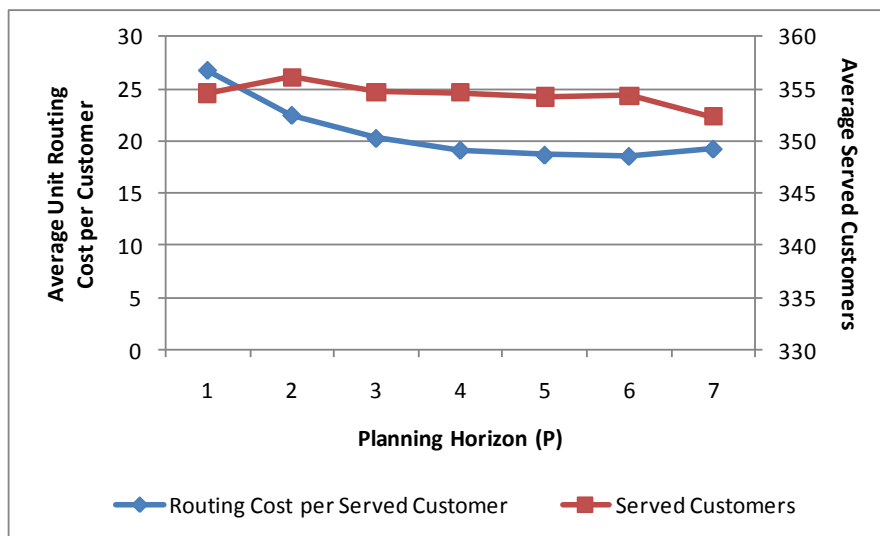
Πίνακας Π.5: Συγκριτικά αποτελέσματα για διαφορετικούς ορίζοντες προγραμματισμού (δυναμική περίπτωση)

Πείραμα	Ορίζοντας Προγραμματισμού			
	P = 3		P = 5	
	Δρομολογημένοι Πελάτες	Κόστος Δρομολόγησης/Πελάτη	Δρομολογημένοι Πελάτες	Κόστος Δρομολόγησης/Πελάτη
L_r103	295	18.80	294	19.14
L_r106	299	17.60	299	18.19
L_r109	299	17.91	299	17.91
L_c106	300	25.09	300	25.14

Πείραμα	Ορίζοντας Προγραμματισμού			
	$P = 3$		$P = 5$	
	Δρομολογημένοι Πελάτες	Κόστος Δρομολόγησης/ Πελάτη	Δρομολογημένοι Πελάτες	Κόστος Δρομολόγησης/ Πελάτη
L_c108	300	24.68	300	24.98
L_c102	299	24.44	299	23.25
L_rc101	290	25.71	283	26.39
L_rc105	296	24.44	294	23.99
L_rc107	300	22.02	300	20.60

Όσον αφορά στα αποτελέσματα που αντιστοιχούν στους δύο ορίζοντες προγραμματισμού  $P$ , δεν διακρίνεται κάποια σημαντική διαφορά, τόσο σε σχέση με τους δρομολογημένους πελάτες, όσο και σε σχέση με το λόγο του κόστους δρομολόγησης ανά πελάτη.

Για να μελετηθεί αναλυτικότερα η επίδραση του ορίζοντα προγραμματισμού, διενεργήσαμε μία σειρά επιπρόσθετων πειραμάτων για τιμές του ορίζοντα προγραμματισμού από 1 έως 7. Στα πειράματα αυτά χρησιμοποιήθηκε μεγαλύτερο εύρος περιοδικής ευελιξίας (υπόδειγμα 7 περιόδων). Για κάθε πείραμα θεωρήθηκαν 360 πελάτες με ρυθμό άφιξης 12 πελατών ανά περίοδο. Στο Σχήμα Π.10 παρουσιάζονται οι μέσες τιμές των αποτελεσμάτων (δρομολογημένοι πελάτες και κόστος δρομολόγησης ανά πελάτη) για όλα τα πειράματα (που αφορούν διαφορετικές γεωγραφικές κατανομές πελατών και διαφορετικό εύρος χρονικών παραθύρων).



Σχήμα Π.10: Αριθμός δρομολογημένων πελατών και κόστος δρομολόγησης ανά πελάτη για τιμές του ορίζοντα προγραμματισμού από  $P = 1, \dots, 7$  (μέσες τιμές για τα πειράματα διαφορετικών γεωγραφικών κατανομών και τιμών του εύρους χρονικών παραθύρων)

Όσον αφορά στον ορίζοντα προγραμματισμού  $P$ , σε όλα τα πειράματα εμφανίζεται μείωση του κόστους δρομολόγησης μέχρι της τιμής  $P = 4$ . Στη συνέχεια, το μέσο κόστος δρομολόγησης ανά πελάτη παραμένει σχεδόν αμετάβλητο παρουσιάζοντας ελαφριά αύξηση για τις τιμές  $P = 6$  and  $7$ ). Επίσης, με βάση το Σχήμα Π.10 δεν παρατηρούνται σημαντικές διαφοροποιήσεις όσον αφορά στο πλήθος των δρομολογημένων πελατών.

## ΜΙΑ ΕΙΔΙΚΗ ΠΕΡΙΠΤΩΣΗ ΠΡΑΚΤΙΚΗΣ ΣΗΜΑΣΙΑΣ

Στην παρούσα διδακτορική διατριβή μελετήθηκε επίσης η περίπτωση στην οποία στόλος οχημάτων εξυπηρετεί δύο είδη πελατών σε περιβάλλον πολλαπλών περιόδων:

- Το πρώτο είδος αφορά πελάτες οι οποίοι έχουν ανατεθεί ήδη σε συγκεκριμένα οχήματα και περιόδους του χρονικού ορίζοντα. Ωστόσο η αλληλουχία των επισκέψεων στους πελάτες αυτούς δεν είναι προκαθορισμένη εντός της διαδρομής κάθε οχήματος. Οι προκαθορισμένοι πελάτες ποικίλουν από περίοδο σε περίοδο.
- Το δεύτερο είδος αφορά πελάτες οι οποίοι παρουσιάζουν περιοδική ευελιξία, γίνονται γνωστοί δυναμικά σε κάθε περίοδο και χαρακτηρίζονται από χρονικά παράθυρα και παράθυρα περιόδων.

Σκοπός του προβλήματος είναι η ελαχιστοποίηση του συνολικού κόστους δρομολόγησης που αφορά στα δύο είδη πελατών. Το πρόβλημα επιλύεται μέσω τεχνικής κυλιόμενου χρονικού ορίζοντα με σκοπό να αντιμετωπιστεί η δυναμική άφιξη των πελατών.

### Απαραίτητες Τροποποιήσεις

Για την επίλυση του προβλήματος αυτού προτείνουμε απαραίτητες αναγκαίες τροποποιήσεις στο μοντέλο και τον τρόπο επίλυσης του ΠΔΟΠΕΧΠ.

#### Τροποποιήσεις Μαθηματικού Μοντέλου

Το μαθηματικό μοντέλο τροποποιείται ώστε να περιλάβει τους προκαθορισμένους πελάτες. Για τον λόγο αυτό ορίζουμε τα σύνολα  $N^m$  (σύνολο των προκαθορισμένων πελατών) και  $N^f$  (σύνολο των γνωστών ευέλικτων πελατών). Σε κάθε προκαθορισμένο πελάτη  $i^r$  ανατίθεται παράθυρο περιόδου  $[\xi_i^s, \xi_i^e] = [\rho_i, \rho_i]$  όπου  $\rho_i$  είναι η περίοδος στην οποία πρέπει να εξυπηρετηθεί ο πελάτης  $i^r$ . Επιπρόσθετα, κάθε πελάτης  $i^r$  πρέπει να εξυπηρετηθεί από συγκεκριμένο όχημα  $r$  που ανήκει στο σύνολο των διαθέσιμων οχημάτων  $K_p = \{k_p^1, \dots, k_p^r, \dots, k_p^{|K_p|}\}$ . Οι περιορισμοί (Π.2) τροποποιούνται σε δύο διακριτούς περιορισμούς ως εξής:

$$\sum_{j \in W} x_{ij\rho_i k_{\rho_i}^r} = 1 \quad \forall i^r \in N_m \quad (\text{Π.23})$$

$$\sum_{p \in I_i} \sum_{k \in K_p} \sum_{j \in W} x_{ijpk} = 1 \quad \forall i \in N_f \quad (\text{Π.24})$$

Οι περιορισμοί (Π.23) ορίζουν ότι κάθε προκαθορισμένος πελάτης πρέπει να εξυπηρετηθεί εντός της συγκεκριμένης περιόδου  $\rho_i$  και από το συγκεκριμένο όχημα  $k_{\rho_i}^r$ , ενώ οι περιορισμοί (Π.24) ορίζουν ότι κάθε ευέλικτος πελάτης πρέπει να εξυπηρετηθεί μία φορά (από ένα όχημα εντός μίας περιόδου του παραθύρου περιόδων  $I_i$ ).

#### Τροποποιήσεις Μεθόδου Δυναμικής Δημιουργίας Μεταβλητών (ΔΔΜ)

Δεδομένης της αρχικής ανάθεσης των προκαθορισμένων πελατών, οι αρχικές «κολώνες» του ΠΚΠ πρέπει να περιλάβουν, τουλάχιστο, τους πελάτες αυτούς. Αρχικοποιούμε την μέθοδο με λύση της οποίας τα αρχικά δρομολόγια περιλαμβάνουν μόνο τους προκαθορισμένους πελάτες. Στη λύση αυτή οι ευέλικτοι πελάτες θεωρούνται ως μη δρομολογημένοι.

Καθώς κάθε όχημα δεσμεύεται από την ύπαρξη των προκαθορισμένων πελατών, η χρήση ενός κοινού ΥΠ για κάθε περίοδο δεν είναι εφικτή.. Συνεπώς, στην περίπτωση αυτή, επιλύεται ξεχωριστό ΥΠ για κάθε συνδυασμό περιόδου και οχήματος.

#### Τροποποιήσεις του ΥΠ

Όπως αναφέρθηκε παραπάνω, κάθε «κολώνα» (δρομολόγιο) πρέπει να περιλαμβάνει όλους τους προκαθορισμένους πελάτες του συνόλου  $N_m^r$ , δηλ. του συνόλου το οποίο αφορά το όχημα  $r$ . Έτσι, η ετικέτα (label)  $L_{\delta i}$  που σχετίζεται με το συγκεκριμένο δρομολόγιο δεν επιτρέπεται να επιστρέψει στο depot εάν δεν έχουν δρομολογηθεί όλοι οι πελάτες του  $N_m^r$ . Επιπρόσθετα, κάθε «κολώνα» πρέπει να μην περιλαμβάνει προκαθορισμένους πελάτες οι οποίοι δε σχετίζονται με το όχημα  $r$ . Η τελευταία απαίτηση αντιμετωπίζεται χρησιμοποιώντας μόνο τους εφικτούς πελάτες σε κάθε ΥΠ για κάθε όχημα  $k_p^r$  (δηλ.  $N_m^r$ ). Σε περίπτωση που ένας πελάτης του  $N_m^r$  δεν μπορεί να εξυπηρετηθεί εντός του σχετικού δρομολογίου, τότε η σχετική ετικέτα απαλείφεται και δεν επεκτείνεται περαιτέρω.

Η ύπαρξη των προκαθορισμένων πελατών εντός κάθε δρομολογίου, απαιτεί την τροποποίηση των κριτηρίων κυριαρχίας (dominance criteria) ώστε όταν συγκρίνεται η ταμπέλα  $L_{\delta i}$  με μία άλλη  $L_{\delta i'}$  να λαμβάνεται υπόψη το πλήθος των πελατών οι οποίοι έχουν ήδη εξυπηρετηθεί από το σχετικό ατελές (partial) δρομολόγιο  $\delta$ . Για το σκοπό αυτό ενδυναμώνουμε τα κριτήρια κυριαρχίας, προσθέτοντας την παράμετρο κόστους  $\bar{c}_{\delta i}$  (κόστος ισοδυναμίας). Το τελευταίο

αντιπροσωπεύει άνω όριο (χειρίστη περίπτωση) του συνολικού κόστους που απαιτείται για να εξυπηρετηθούν όλοι οι προκαθορισμένοι πελάτες, οι οποίοι δεν έχουν ακόμα συμπεριληφθεί εντός του ατελούς δρομολογίου δ.

### Πειραματική Διερεύνηση Κυλιόμενου Χρονικού Ορίζοντα με Προ-ανατεθειμένους Πελάτες

Στην διερεύνηση της παρούσας περίπτωσης χρησιμοποιήθηκαν τα πειραματικά δεδομένα που περιγράφηκαν στην προηγούμενη Παράγραφο, ήτοι:

- Χρησιμοποιήθηκαν τρία διαφορετικά υποδείγματα περιόδων (3, 5 και 7 περιόδων) .
- Ως προκαθορισμένοι πελάτες επιλέχθηκαν τυχαία 180 πελάτες. Για τους πελάτες αυτούς ορίστηκε το μεγαλύτερο δυνατό χρονικό παράθυρο
- Ορίστηκε ρυθμός άφιξης 6 δυναμικών πελατών σε κάθε περίοδο
- Επιλέχθηκαν τρία πειράματα, ένα για κάθε γεωγραφική κατανομή πελατών (R1, C1 και RC1)
- Τέλος, δύο οχήματα θεωρήθηκαν διαθέσιμα σε κάθε περίοδο με μέγιστο αριθμό προκαθορισμένων πελατών ανά όχημα ίσο με τρία.

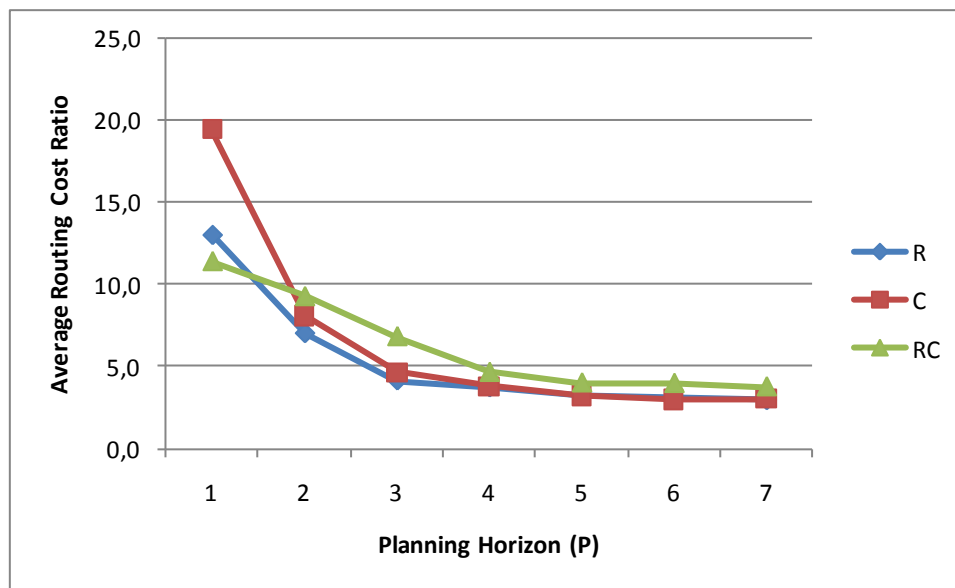
Ο Πίνακας Π.6 παρουσιάζει τα αποτελέσματα ανά γεωγραφική κατανομή πελατών, παράθυρο περιόδων και ορίζοντα προγραμματισμού. Τα στοιχεία που παρουσιάζονται στον Πίνακα είναι τα εξής: (α) Ο μέσος αριθμός δυναμικών πελατών που δρομολογήθηκαν στον μακροχρόνιο ορίζοντα  $S$ . (β) Το μέσο επιπρόσθετο κόστος ανά δυναμικό πελάτη. Το επιπρόσθετο κόστος υπολογίζεται αφαιρώντας από το συνολικό κόστος δρομολόγησης κάθε περιόδου, το αρχικό κόστος δρομολόγησης των προκαθορισμένων πελατών.

Πίνακας Π.6: Μέσες τιμές αποτελεσμάτων ανά τύπο προβλήματος, υπόδειγμα παραθύρου περιόδων και ορίζοντα προγραμματισμού

P	Τύποι Προβλήματος (Γεωγραφική Κατανομή)					
	R1		C1		RC1	
	Δρομολογημένοι Πελάτες	Κόστος Δρομολόγησης/ Πελάτη	Δρομολογημένοι Πελάτες	Κόστος Δρομολόγησης/ Πελάτη	Δρομολογημένοι Πελάτες	Κόστος Δρομολόγησης/ Πελάτη
Υπόδειγμα Παραθύρου Περιόδων 3						
1	176.0	13.8	180.0	19.6	162.3	12.0
2	177.0	8.2	180.0	9.5	164.3	11.0
3	177.0	7.0	180.0	8.0	163.0	10.2
Υπόδειγμα Παραθύρου Περιόδων 5						
1	177.3	13.1	180.0	19.5	174.7	11.9
2	177.3	7.1	180.0	8.3	176.3	9.5
3	177.3	4.4	180.0	5.2	176.3	8.2

P	Τύποι Προβλήματος (Γεωγραφική Κατανομή)					
	R1		C1		RC1	
	Δρομολογημένοι Πελάτες	Κόστος Δρομολόγησης/ Πελάτη	Δρομολογημένοι Πελάτες	Κόστος Δρομολόγησης/ Πελάτη	Δρομολογημένοι Πελάτες	Κόστος Δρομολόγησης/ Πελάτη
4	177.3	4.1	180.0	4.1	176.0	5.7
5	177.3	4.1	180.0	4.2	175.7	6.4
Υπόδειγμα Παραθύρου Περιόδων 7						
1	177.3	13.0	180.0	19.5	174.3	11.4
2	177.3	7.0	180.0	8.1	176.3	9.3
3	177.3	4.1	180.0	4.6	176.7	6.8
4	177.3	3.7	180.0	3.8	176.3	4.6
5	177.3	3.2	180.0	3.2	175.3	4.0
6	177.0	3.1	179.7	3.0	175.3	4.0
7	177.0	2.9	179.7	3.0	175.3	3.8

Στο Σχήμα Π.11 παρουσιάζεται ενδεικτικά ο μέσος λόγος του κόστους δρομολόγησης ανά πελάτη για το υπόδειγμα παραθύρου περιόδων 7 και για όλους τους ορίζοντες προγραμματισμού. Παρόμοια αποτελέσματα ελήφθησαν και για τα άλλα υποδείγματα παραθύρου περιόδων (δηλ. 3 και 5).

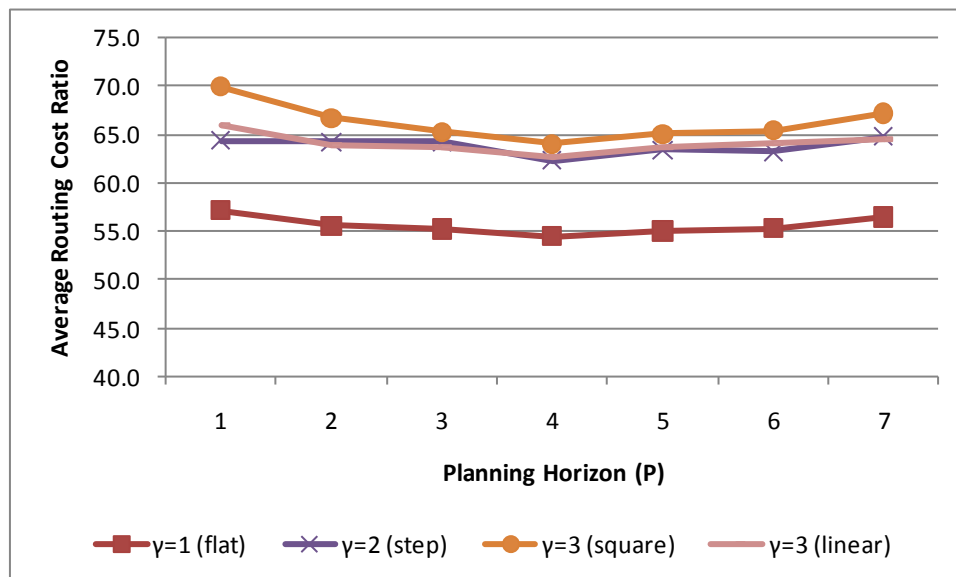


Σχήμα Π.11: Μέσο κόστος δρομολόγησης ανά τύπο προβλήματος (R1, C1, RC1) για το υπόδειγμα 7

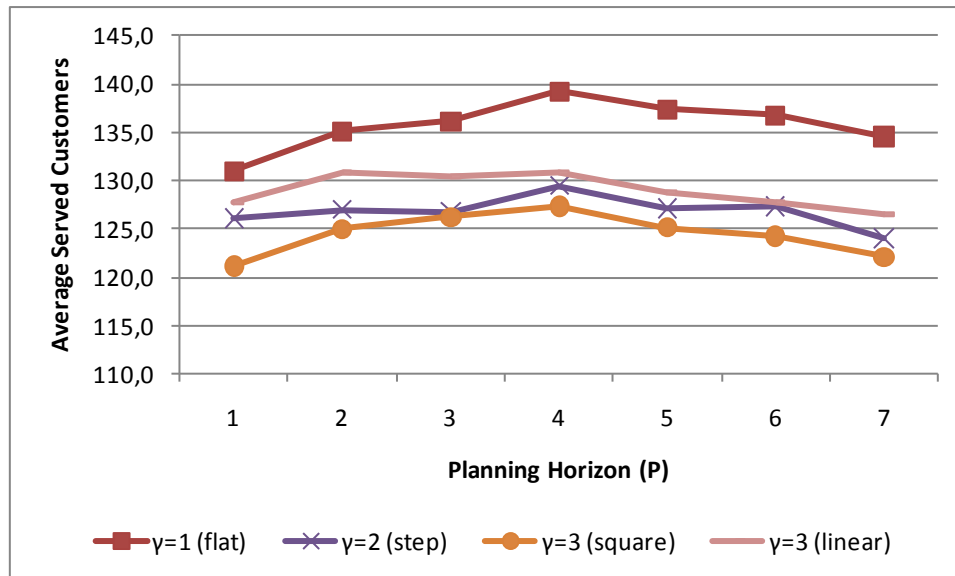
Σε όλους τους τύπους προβλημάτων και ειδικότερα για τα υποδείγματα 5 και 7, το κόστος δρομολόγησης ανά πελάτη μειώνεται σημαντικά για τις αρχικές τιμές του  $P$ . Η μείωση αυτή εμφανίζει σταθεροποίηση μετά από συγκεκριμένη τιμή του  $P$  (π.χ. για  $P = 4$  στο υπόδειγμα 7). Όσον αφορά στον τύπο προβλήματος (R1, C1, RC1), η μείωση αυτή είναι πιο εμφανής για τα προβλήματα R1 και τα C1, ενώ στα προβλήματα RC1 παρουσιάζεται μετριασμένη τάση μείωσης του κόστους.

### Ενδεικτικά Πειραματικά Αποτελέσματα για την Περίπτωση Μεγάλου Αριθμού Μη-Δρομολογημένων Πελατών

Διερευνήθηκε, επίσης, η αποτελεσματικότητα των προτεινόμενων μεθόδων σε περιπτώσεις όπου μόνο ένα περιορισμένο μέρος του συνόλου των δυναμικών πελατών μπορεί να δρομολογηθεί λόγω αυστηρών περιορισμών πόρων. Για τον λόγο αυτό χρησιμοποιήθηκε το υπόδειγμα περιόδων 7 και αυξήθηκε ο χρόνος εξυπηρέτησης κάθε δυναμικού πελάτη κατά 100%. Οι υπόλοιπες παράμετροι παραμένουν όπως στην προηγούμενη πειραματική διερεύνηση. Μελετήθηκαν διαφορετικές συναρτήσεις ποινής, επιπλέον της γραμμικής ( $\gamma = 5$ ):  $\gamma = 1$  (ομοιόμορφη),  $\gamma = 2$  (βηματική), και  $\gamma = 3$  (τετραγωνική). Τα Σχήματα Π.12 και Π.13 παρουσιάζουν τα αποτελέσματα για το μέσο λόγο κόστους δρομολόγησης ανά ορίζοντα προγραμματισμού και το μέσο αριθμό δρομολογημένων δυναμικών πελατών.



Σχήμα Π.12: Μέσος λόγος κόστους δρομολόγησης ανά ορίζοντα προγραμματισμού και συνάρτηση ποινής (για όλα τα πειράματα) – Υπόδειγμα 7



Σχήμα Π.12: Μέσος αριθμός δρομολογημένων δυναμικών πελατών ανά ορίζοντα προγραμματισμού και συνάρτηση ποινής (για όλα τα πειράματα) – Υπόδειγμα 7

Η ομοιόμορφη συνάρτηση  $\gamma = 1$ , στην οποία όλοι οι δυναμικοί πελάτες έχουν της ίδια ποινή, επιτυγχάνει τα ευνοϊκότερα αποτελέσματα, όσον αφορά στο κόστος δρομολόγησης αλλά και στον αριθμό των δρομολογημένων πελατών. Το αποτέλεσμα αυτό αποδίδεται στα εξής: (α) Η ομοιόμορφη συνάρτηση δεν εμπλέκεται με την διαδικασία δρομολόγησης μη αποδίδοντας επιλεκτική προτεραιότητα σε πελάτες, και (β) πολλοί δυναμικοί πελάτες δε δρομολογούνται σε κάθε περίπτωση.

Όσον αφορά στον ορίζοντα προγραμματισμού, οι ενδιάμεσοι ορίζοντες επιτυγχάνουν βελτιωμένα αποτελέσματα σχετικά με το πλήθος των δρομολογημένων πελατών, καθώς και περιορισμένη βελτίωση στο κόστος δρομολόγησης.



## ABSTRACT

In this dissertation we investigate the Multi-Period Vehicle Routing Problem with Time Windows (MPVRPTW), in which customer orders are related to a *period window* (a set of service periods). Routing costs are minimized over a planning horizon, respecting period window, time window, and capacity constraints. We present a general model and an exact approach to solve this problem based on the column generation method. We also propose two novel, efficient techniques to speed up the column generation method for obtaining lower bounds. The proposed techniques exploit the multi-period setting in order to identify similarities within the subproblems and, thus, avoid solving all subproblems at each iteration. We evaluated the performance of the proposed methods systematically for various parameters, such as customer geographical distribution and period window patterns. In most cases, the new methods improve significantly the efficiency of convergence to the optimal solution of the relaxed problem, especially in the computationally expensive test cases with wide period windows.

Integer optimal solutions to the MPVRPTW are provided through a branch-and-price implementation. We propose two different strategies that consider the multi-period characteristics of the problem, in addition to a simple pruning heuristic that speeds up the solution procedure and provides efficient results.

For solving the MPVRPTW in long-term horizons, we propose a rolling horizon framework. Initially, we discuss three theoretical statements that provide insights on the effects of the planning and implementation horizons in the final solutions. Subsequently, in order to apply rolling horizon routing, we propose significant modifications to the model and the solution approach for the MPVRP; these modifications concern the ability to postpone serving customers for later periods. We investigate two rolling horizon settings (quasi-static and dynamic) and we establish the recommended values for the planning and implementation horizons, under a wide range of parameters, such as customer geographical distribution and time window width.

Finally, we address a practical variation, which regards a hybrid service policy that includes (a) inflexible (pre-assigned to specific vehicles) and (b) flexible customer orders. For this case, we propose the necessary modifications to the MPVRP model and solution approach. Extensive experiments show that significant cost savings can be achieved by considering longer planning horizons in the planning process.



## CONTENTS

<b>Chapter 1: Introduction .....</b>	<b>1</b>
<b>Chapter 2: Theoretical Background .....</b>	<b>7</b>
2.1 Related Problems in the Literature .....	7
2.2 Branch and Price through Column Generation .....	19
2.3 Research Contribution .....	22
<b>Chapter 3: The Multi-Period Vehicle Routing Problem with Time Windows .....</b>	<b>25</b>
3.1 Mathematical Formulation .....	25
3.2 Lower Bounds through Column Generation .....	28
3.3 The ESPPTWCC – Solution Procedure .....	35
3.4 Combining RMP with the Subproblems.....	42
<b>Chapter 4: Accelerating Techniques for the MPVRPTW .....</b>	<b>45</b>
4.1 Unified Subproblem Method.....	46
4.2 Cloning Method.....	50
4.3 Implementation of Parallel Solution.....	53
4.4 Computational Techniques for Solving the Subproblems.....	53
4.5 Test Instances and Benchmark Results .....	57
<b>Chapter 5: Branch and Price: Obtaining Integer Solutions .....</b>	<b>71</b>
5.1 The Generic Branch and Price for the VRPTW .....	71
5.2 Branch and Price for the Multi-Period Routing Problem.....	74
5.3 Heuristic Pruning for Branch and Price.....	78
5.4 Branch and Price Testing.....	80
<b>Chapter 6: Enhancements for Applying the MPVRPTW in a Rolling Horizon Framework.....</b>	<b>89</b>
6.1 The Rolling Horizon Planning Process .....	89
6.2 Theoretical Insights for a Special Case of Rolling Horizon Planning.....	91
6.3 Modifying MPVRPTW to Deal with Limited Resources and Unserved Customers ...	97
6.4 Experimental Investigation of Rolling Horizon Routing .....	107
<b>Chapter 7: The MPVRPTW with Pre-Assigned Customers .....</b>	<b>119</b>

7.1 Practical Applications of the MPVRP with Pre-Assigned Customers .....	120
7.2 Modifications for the Multi-Period Routing Problem with Pre-Assigned Customers	122
7.3 Experimental Investigation of Rolling Horizon with Pre-Assigned Customers .....	126
<b>Chapter 8: Conclusions and Future Research .....</b>	<b>135</b>
8.1 Conclusions .....	135
8.2 Future Research .....	139
<b>References .....</b>	<b>141</b>
<b>Appendices .....</b>	<b>153</b>
Appendix A: Lower Bounds and Computational Times for all CG Techniques .....	153
Appendix B: Branch-And-Price Results .....	159
Appendix C: Detailed Results on the Experiments Regarding the Penalty Functions.....	167
Appendix D: Detailed Results of the Experimental Investigation presented in Chapter 6	172
Appendix E: Detailed Results of the Experimental Investigation Presented in Chapter 7	179

## LIST OF FIGURES

Figure 2.1: Evolution of models and solution methods for the PVRP (Francis <i>et al.</i> , 2008) ..	15
Figure 2.2: Related problems with the multi period routing problem.....	17
Figure 3.1: Master Problem (MP) and Restricted Master Problem (RMP). Set partitioning formulation. ....	31
Figure 3.2: Column generation procedure for multi-period problems .....	43
Figure 4.1: Common routes ratio for different sizes of set $N_1$ and $N_1 2$ .....	46
Figure 4.2: Feasible periods per Customer.....	50
Figure 4.3: Illustrative example of the Cloning strategy suboptimality .....	52
Figure 4.4: Sequential procedure of the classical column generation and the parallel approaches for the Multi-period problem.....	53
Figure 4.5: Pseudocode for label extension using looping procedures .....	54
Figure 4.6: Set $B(i)$ representation in a 2-D matrix. ....	54
Figure 4.7: Set $S(i)$ representation in a 2-D matrix. ....	55
Figure 4.8: Time windows of R101 – R106 instances .....	58
Figure 4.9: Time windows of the R107 – R112 instances .....	58
Figure 4.10: Period window patterns .....	60
Figure 4.11: Cumulative computational time per method and TW category.....	64
Figure 4.12: Computational time (% difference from the classical method) .....	65
Figure 4.13: Cumulative computational times per problem set and method (hrs).....	66
Figure 4.14: Normalized cumulative computational times of alternative methods with respect to the classical approach (100%) .....	67
Figure 4.15: Cumulative computational times per pattern for the R1 Instances.....	67
Figure 4.16: Cumulative computational times per pattern for the C1 Instances.....	68
Figure 4.17: Cumulative computational times per pattern for the RC1 Instances .....	69
Figure 5.1: Branch and Price procedure.....	72

Figure 5.2: Multi period Branch and Price techniques.....	76
Figure 5.3: Example of the heuristic pruning technique .....	80
Figure 5.4: Average computational time ratio (CLONE vs. FULL) .....	84
Figure 5.5: Nodes explored per period window pattern.....	85
Figure 5.6: Computational times and deviation from optimal integer solution for different values of $\lambda$ ( $P + 1$ method).....	87
Figure 6.1. Planning process .....	90
Figure 6.2: Customers and related network (example 2) .....	94
Figure 6.3: Customers and routes for implementation horizons $M = 1$ and $2$ (Example 3)...	96
Figure 6.4: Penalty functions ( $\gamma = 1, \dots, 5$ ).....	104
Figure 6.5: Ratio of unserved customers (%).....	106
Figure 6.6: Cost per served customer .....	106
Figure 6.7: Unnerved customers and cost per served customer (all five penalty functions) .	106
Figure 6.8: Cumulative unit cost per routed customer and period (L_R109 Instance) .....	112
Figure 6.9: Average cost per routed customer (per problem type) .....	115
Figure 6.10: Routing cost per served customer and served customers per planning horizon (average over all instances and types) .....	116
Figure 7.1: Average routing cost ratio per problem type (Pattern 3) .....	128
Figure 7.2: Average routing cost ratio per problem type (Pattern 5) .....	128
Figure 7.3: Average routing cost ratio per problem type (Pattern 7) .....	129
Figure 7.4: Routing cost ratio and served customers per planning horizon (average over all instances) – Pattern 3 .....	130
Figure 7.5: Routing cost ratio and served customers per planning horizon (average over all instances) – Pattern 5 .....	130
Figure 7.6: Routing cost ratio and served customers per planning horizon (average over all instances) – Pattern 7 .....	130

Figure 7.7: Routing cost ratio and served customers per planning horizon (average over all instances) – Pattern 7 .....	132
Figure 7.8: Average routing cost ratio per planning horizon and penalty function (average over all instances) – Pattern 7 .....	133
Figure 7.9: Average served customers per planning horizon and penalty function (average over all instances) – Pattern 7 .....	133

**LIST OF TABLES**

Table 2.1: Multi-Period Vehicle Routing Problems.....	17
Table 3.1: Improvement Procedures for (E)SPPTWCC .....	37
Table 4.1: Average time windows per test instance (maximum allowable time: 230, 1236 and 240 for the R1, C1 and RC1 instances, respectively).....	59
Table 4.2: Period-window patterns .....	60
Table 4.3: Comparison of computational time of strategies (% difference from the FULL method).....	65
Table 4.4: Computational times per problem set (hrs).....	66
Table 4.5: Comparison of alternative methods per factor .....	69
Table 5.1: Validation of the branch and price algorithm using Solomon benchmarks .....	81
Table 5.2: Aggregate B&P results for the 66 instances solved by both techniques ( <b>2br</b> and <b>P + 1</b> ).....	82
Table 5.3: Instances with an integer solution using the FULL or CLONE method.....	83
Table 5.4: Performance of pruning heuristic for different values of threshold $\lambda$ using the <b>P + 1</b> branching method (66 instances solved) .....	85
Table 5.5: Performance of pruning heuristic for different values of threshold $\lambda$ using the <b>2br</b> branching method (66 instances solved) .....	86
Table 6.1: Customers and related period windows (example 2).....	94
Table 6.2: Costs of arcs (example 2).....	94
Table 6.3: Final routes per period for implementation horizon of <b>P = 2</b> and <b>3</b> (example 2) .	94
Table 6.4: Customers and related period windows (example 3) .....	95
Table 6.5: Costs of arcs (example 3).....	96
Table 6.6: Routes and cost per period for implementation horizons <b>M = 1</b> and <b>2</b> (Example 3) .....	96
Table 6.7: Average test results for 16 test instances per penalty function and planning horizon .....	105

Table 6.8: Comparative results for different planning and implementation horizons (quasi-static case) .....	110
Table 6.9: Paired difference t-test for factor <b>P</b> (2 levels) - $\alpha = 0,05$ .....	111
Table 6.10: Paired difference t-test for factor <b>M</b> (2 levels), $\alpha = 0,05$ .....	111
Table 6.11: Example illustrating the effect of the implementation horizon in the dynamic rolling horizon case .....	113
Table 6.12: Comparative results using different planning horizons (dynamic arrival of customers).....	113
Table 6.13: Paired difference t-test for factor <b>P</b> (2 levels) .....	114
Table 6.14: Average results per planning horizon and problem type (dynamic arrival of customers).....	114
Table 6.15: ANOVA for factor <b>P</b> varying from 1 to 7 (7 levels) w.r.t. cost per served customer .....	116
Table 6.16: ANOVA for factor <b>P</b> varying from 4 to 7 (4 levels) w.r.t. cost per served customer .....	117
Table 6.17: ANOVA Test for factor <b>P</b> (7 levels) w.r.t. Served Customers .....	117
Table 7.1: Average experimental results per problem type, period window pattern, and planning horizon.....	127
Table 7.2: Paired difference t-test analysis for factor <b>P</b> and routing cost ratio .....	131



## Chapter 1: INTRODUCTION

During the last five decades, transportation and distribution of goods have received considerable attention from both industry and academia. The major focus of this interest has been to simultaneously minimize logistics costs and maximize service quality. In order to obtain efficient solutions to related practical problems, the research community has used the fundamentals of Operations Research (OR) to pose basic problems, construct robust models, and develop effective approaches based on both heuristics and exact methods of integer programming.

One of the initial problems that received considerable research interest is the Travelling Salesman Problem (TSP), which has set the basis for significant subsequent work in this field. In the TSP, a single vehicle (or salesperson) is tasked to visit a set of customers using the minimum cost route. This problem has been shown to be NP-hard (Garey and Johnson, 1979); however a wealth of solution procedures and methods has been developed to address it effectively, based on network theory, linear programming and other approaches, such as heuristics and metaheuristics.

A second fundamental problem of equivalent contribution in the area is the so called Vehicle Routing Problem (VRP), which targets the design of a set of minimum cost routes, each served by a vehicle belonging to a fleet, starting and ending at a depot and serving a set of customers with known demand and service costs. These routes are designed subject to several constraints, such as limited total time of travel (route length), or limited vehicle capacity. Numerous variations of this problem exist, using different objectives and constraints, depending on the problem under investigation.

The majority of methods and systems used in practice for vehicle routing deal with single-period problems under known demand; that is, orders of known demand are provided for a certain period (e.g. day) and are serviced within this period. Although in various practical cases this setting is appropriate, there are many other cases in which the orders can be served within a *period window*, i.e., a consecutive set of periods (days). This simple alteration complicates the routing procedure by adding another critical factor: The selection of customers (orders) to be served in each period, in order to minimize the total cost (or distance) for the entire time horizon.

Related problems are faced in practice by appointment-based logistics systems, such as those offering on-site repair/maintenance services, home delivery of products, or hybrid courier services that perform both next day and bulk deliveries. These problems are typically dealt by a two-phase procedure: In the first phase, the customer orders are allocated/assigned in a specific period within the planning horizon using simple rules (e.g. First-Come-First-Serve or geographical grouping) without considering the routing costs directly. In the second phase, the vehicle routing problem is solved for each period of the horizon in order to minimize routing costs. This practice, however, may lead to suboptimalities, since routing costs are not considered simultaneously with the allocation of orders in the periods of the planning horizon. This latter consideration is the focus of the Multi-Period Vehicle Routing Problem.

### **The Multi-Period Vehicle Routing Problem with Time Windows (MPVRPTW)**

Consider a dedicated fleet of vehicles (of fixed capacity) that is available to serve customer orders starting operations from a single depot. Each customer order is related to a period window, which depends on the service level provided to that customer. The period window is a set of consecutive periods. In addition to the period window, each order may be related to a time window. The latter concerns the allowable time interval within each period for delivering the service. Practical examples concerning the time window include the case in which it may not be feasible to provide service to a customer outside typical working hours (i.e. early in the morning or late at night), or the case in which individual customers are available for service only after the end of a business day (e.g. after 17:00). In addition to the above characteristics, the MPVRPTW addresses simultaneously a set of consecutive periods (planning horizon); that is, all known customers with period window which starts or ends within the planning horizon are considered for assignment in the appropriate period and vehicle, and are also planned/ routed targeting efficiency.

In the current dissertation we study the above problem and propose two new exact strategies to provide efficient lower bounds to the MPVRPTW. To do so we take advantage of the special structure of the multi-period problem. We validate the efficiency gains by comparing against benchmarks used by other approaches. Additionally, we develop schemes to obtain the integer optimal solution, which are relevant to the multi-period setting. We also propose a simple pruning heuristic in order to accelerate the solution procedure. The latter is suitable for multi-period vehicle routing problems that are solved over long horizons using a rolling horizon framework.

## **Rolling horizon routing**

We also address vehicle routing problems in long-term horizons (say of length  $S$ ). For this setting, we propose a rolling horizon framework in which the solution procedure of the MPVRPTW is embedded. In this framework, initially we solve a single MPVRPTW in a planning horizon of selected length (say  $P$ ) by taking under consideration all known customer orders with period window that starts or ends within this horizon. Based on the resulting solution, the assignments and routes of the first  $M$  periods of the horizon are implemented. The remaining customer orders, along with newly arriving ones, are considered for the next planning horizon by solving again a new MPVRPTW. These steps are repeated until the long-term horizon is exhausted.

Considering this rolling horizon framework, we address two distinct cases:

The **quasi-static** case, in which all customer orders within the long term planning horizon  $S$  are considered to be known. When using a rolling horizon scheme with planning horizon of length  $P$  and implementation horizon of length  $M$ , the solution cycle will be repeated every  $M$  periods. In this case, each time we solve the MPVRPTW for  $P$  periods, the only new customer orders considered are those of the last  $M$  periods of this planning horizon.

The **dynamic** case, in which new customer orders arrive during every period. In this case, not all orders to be served within the planning horizon are known in advance. However, there is full knowledge of the customer orders of the next period to be planned.

To apply rolling horizon routing, significant modifications are proposed to both the model and the solution approach of the MPVRP; these modifications concern the ability to postpone serving clients from one planning horizon to the next. Based on these modifications, the current dissertation investigates in depth the effects of the two critical parameters of the rolling horizon scheme: the implementation horizon ( $M$ ) and the planning horizon ( $P$ ). Theoretical principles are established for the quasi-static case. These principles, and other significant insights, are then studied through an extensive experimental investigation conducted for both the quasi-static and the dynamic cases mentioned above.

## **A special case of practical significance**

This dissertation also addresses a special case of rolling horizon routing, which we have encountered in practice. In this case, some customer orders have been pre-assigned to periods and vehicles (inflexible orders), while some others arrive dynamically and may be assigned to

any period of the planning horizon, within, of course, their period window (flexible orders). We propose the appropriate modifications to the MPVRP model and the solution approach. Furthermore, an extensive experimental investigation is conducted in order to obtain significant insights in the parameters of the rolling horizon scheme and in the solution approach.

### **Structure of the Dissertation**

The remainder of the dissertation is organized as follows:

Chapter 2 presents and discusses the related problems in the literature, such as the Vehicle Routing Problem with Time Windows (VRPTW), the Periodic Vehicle Routing Problem (PVRP), the Inventory Routing Problem (IRP) and the Multi-Period Vehicle Routing Problem (MPVRP). The similarities and differences of these problems with respect to the problems studied in this dissertation are discussed. Furthermore, Chapter 2 reviews significant literature on the column generation method, identifies research gaps, and highlights the contributions of this dissertation.

Chapter 3 describes the mathematical formulation of the MPVRPTW. It also presents the remodeling of the problem into a framework amenable to column generation. Significant background technical information regarding the column generation method is provided.

Chapter 4 explores new alternative column generation techniques that target improved computational times by exploiting the multi-period structure of the problem. In order to evaluate these techniques, new test instances have been developed based on the Solomon benchmarks with different patterns regarding the period windows of the customer orders. Conclusions are presented based on the analysis of the test results.

Chapter 5 presents the Branch and Price (B&P) framework for the MPVRPTW. Initially, the generic B&P techniques for the VRPTW are presented. These techniques are extended in order to take into consideration the multi-period aspect of the current problem. Two new B&P techniques are presented along with a heuristic that provides solutions in an efficient manner both in terms of the computational time and the value of the objective function.

Chapter 6 provides a formal description of the MPVRPTW within a rolling horizon framework. It includes several enhancements developed to apply the MPVRPTW in cases with limited resources. We study both the quasi-static and the dynamic MPVRP. For the first case, we propose and discuss three theoretical statements concerning the implementation horizon  $M$  and the planning horizon  $P$ . Subsequently, we investigate experimentally the

effects of  $P$  and  $M$  on the quality of the solutions obtained for a large range of experimental cases and input parameters.

Chapter 7 investigates the problem variation concerning the mix of flexible and inflexible orders. This problem is solved on a rolling horizon using appropriate modifications in the column generation scheme. The efficiency of the proposed method is validated through an extensive experimental study, which indicates that significant cost savings can be achieved by considering wider planning horizons in the planning process.

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



## **Chapter 2: THEORETICAL BACKGROUND**

As already mentioned in Chapter 1, this dissertation focuses on long horizon routing problems. These fall into the general category of vehicle routing problems (VRPs) and to variations, such as the Periodic and the Inventory Routing Problems. The dissertation focuses on two major classes of problems: (a) The multi-period routing problem, and b) long horizon routing problems that are solved using the multi-period problem in a rolling horizon approach (both in its quasi-static and dynamic form).

The related theoretical background is discussed in Section 2.1 below. Section 2.2 provides a targeted discussion on the essentials of the basic techniques employed; i.e. column generation, elementary shortest path with resource constraints and Branch-and-Price, respectively. These techniques form the foundation of the new methods proposed to derive exact solutions for the aforementioned problems. Finally, Section 2.3 highlights the contributions of the dissertation in the area of multi-period routing problems.

### **2.1 RELATED PROBLEMS IN THE LITERATURE**

The following review focuses on topics related to the current work: Section 2.1.1 overviews the VRPTW which forms the basis of the multi-period routing problem. Section 2.1.2 presents significant periodic routing problems found in the literature. Section 2.1.3 drills down into the area of multi-period routing problems, which are highly related to the dissertation topic, and highlights the related similarities and differences.

#### **2.1.1 THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS**

The Vehicle Routing Problems (VRP) is one of the most studied problems in both Operational Research and Logistics and it is related to many theoretical and practical transportation problems (Clarke and Wright, 1964; Golden and Assad, 1998; Laporte and Osman, 1995). The VRP falls into the general category of network optimization problems, and is a generalization of the classic Traveling Salesman Problem (TSP) (Christofides, 1979; Cornuejols and Nemhauser, 1978; Gendreau *et al.*, 1997). Specifically, the VRP consists of finding a set of routes to serve a number of geographically dispersed customers at minimum cost. It was introduced by Dantzig and Ramser (1959), who proposed the mathematical formulation and a solution approach for a practical problem of gasoline delivery to service

stations. Since then, a large number of researchers have introduced various theoretical and practical aspects into the related mathematical models.

The VRP is an NP-hard problem (Lenstra and Kan, 1981) and, therefore, practical (large) problem instances cannot be solved to optimality within reasonable time. An insightful survey of significant results in VRP-related research is given by Toth and Vigo (2002). Additionally, the latest advances in VRP research, problem variations, significant methodological approaches, and practical applications are presented in Golden *et al.* (2008). It is noted that the VRP usually refers to its most common variation, the Capacitated VRP (CVRP) (Toth and Vigo, 2002). In CVRP the customer demand is deterministic, and the available fleet is considered to be homogeneous with vehicles of a certain capacity. All vehicles start from the same depot and each customer should be serviced within a single visit (i.e. multiple visits are not allowed). The scope of the problem is to create the least cost routes that serve all customers and respect capacity constraints. Other well-known variations of the VRP are the following:

- VRP with Time Windows (Cordeau *et al.*, 2002)
- VRP with Pickup and Delivery (Toth and Vigo, 2002; Daganzo and Hall, 1993)
- Distance Constrained VRP (Toth and Vigo, 2002)
- Multi-Depot VRP (Bianco *et al.*, 1994; Carpaneto *et al.*, 1989)
- Heterogeneous Capacitated VRP (Taillard, 1996)
- VRP with Backhauls (Toth and Vigo, 2002; Golden *et al.*, 1988 )
- Periodic VRP (Tan and Beasley, 1984; Christofides and Beasley, 1984)

The Vehicle Routing Problem with Time Windows (VRPTW), in addition to the constraints of the CVRP, requires that customers are served within a short time period (time window). Note that usually only the start of the service is required to be included in the time window. Furthermore, a maximum vehicle travel time is specified, which is an upper limit of the total time each vehicle can operate. As stated in Larsen (2001):

*"The VRPTW contains several NP-Hard optimization problems implying that VRPTW is also NP-Hard. Among the NP-Hard problems contained as special cases are TSP (Garey and Johnson, 1979; Lenstra and Kan, 1981), Bin Packing (Garey and Johnson, 1979) and VRP (Lenstra and Kan, 1981)."*

The seminal work by Cordeau *et al.* (2002) provides a comprehensive description of the problem and of the related solution approaches. Since the VRPTW is the basis of the

problems studied here, the related mathematical formulation by Cordeau *et al.* (2002) is discussed subsequently.

Consider a directed graph  $G = (V, A)$ , a set of customers  $N = \{1, 2, \dots, n\}$  and a fleet of homogeneous vehicles  $V$ . The graph contains  $|N| + 2$  vertices, that is,  $|N|$  customers plus the two starting and ending positions of the vehicle fleet; let vertices 0 and  $n + 1$  denote these positions, respectively. The entire set of vertices  $\{0, 1, \dots, n + 1\}$  is denoted as  $C$ . The set of arcs  $A$  represents the direct connections among all vertices, including the starting and ending positions. Each arc  $(i, j)$ , where  $i \neq j$ , has an associated cost  $c_{ij}$  and a travel time  $t_{ij}$ . All vehicles have the same capacity  $q$  and each customer is associated with a demand  $d_i$ . Each customer  $i$  must be served within a certain time window  $[a_i, b_i]$ . In case a vehicle arrives before the opening time of a time window, it must wait until  $a_i$  to start serving the corresponding customer, while service cannot be provided in case the vehicle arrives after the ending time  $b_i$ . Finally, each customer is associated with a deterministic service time  $st_i$ , which, for simplicity and without loss of generality, is incorporated in the traversing time  $t_{ij}$  of the corresponding arcs.

The model presented below contains two sets of decision variables,  $x$  and  $s$ . The variable  $s_{ik}$  for each vertex  $i$  denotes the time when vehicle  $k$  starts to serve customer  $i$ . For each arc  $(i, j)$ , where  $i \neq j, i \neq n + 1, j \neq 0$  and each vehicle  $k$ , the variables  $x_{ijk}$  are defined as:

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ traverses arc } (i, j) \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

The mathematical formulation of the VRPTW is:

$$\min \sum_{k \in V} \sum_{i \in C} \sum_{j \in C} c_{ij} x_{ijk} \quad (2.2)$$

$$\text{s.t.} \quad \sum_{k \in V} \sum_{j \in C} x_{ijk} = 1 \quad \forall i \in N \quad (2.3)$$

$$\sum_{j \in C} x_{0jk} = 1 \quad \forall k \in V \quad (2.4)$$

$$\sum_{i \in C} x_{ihk} - \sum_{j \in C} x_{hjk} = 0 \quad \forall h \in N, \forall k \in V \quad (2.5)$$

$$\sum_{i \in C} x_{i,n+1,k} = 1 \quad \forall k \in V \quad (2.6)$$

$$\sum_{i \in N} d_i \sum_{j \in C} x_{ijk} \leq q \quad \forall k \in V \quad (2.7)$$

$$s_{ik} + t_{ij} - M(1 - x_{ijk}) \leq s_{jk} \quad \forall i, j \in C, \forall k \in V \quad (2.8)$$

$$a_i \leq s_{ik} \leq b_i \quad \forall i \in C, \forall k \in V \quad (2.9)$$

$$x_{ijk} \in \{0,1\} \quad \forall i, j \in C, \forall k \in V \quad (2.10)$$

The objective function (2.2) represents the actual cost of traversing the network's arcs by the available vehicles  $V$ . Constraint (2.3) ensures that each customer is visited by a single vehicle and exactly once. Constraints (2.4) and (2.6) state that each vehicle will start and end at the depot. Constraint (2.5) is the flow conservation constraint, that is, if a vehicle serves a customer, it is also required to depart from that customer. Constraint (2.7) relates to the vehicles capacity  $q$ . Constraint (2.8) ensures that customer  $j$  is served after  $s_{ik} + t_{ij}$ . Note that  $M$  represents a large number and it is used in order to linearize the non-linear constraint  $x_{ijk}(s_{ik} + t_{ij}) \leq x_{ijk}s_{jk}, \forall i, j \in C, \forall k \in V$ . Inequalities (2.9) represent the time window constraints, and relationship (2.10) represents the binary conditions for the problem variables.

The methods available to solve many types of the VRP can be grouped into three main classes (Cordeau *et al.*, 2002): (a) Heuristic algorithms, (b) metaheuristics, and (c) exact approaches. Cordeau *et al.* (2002) present an extended review of the aforementioned classes, while Gendreau *et al.* (2008) present a comprehensive review of metaheuristics for several VRP variations including the VRPTW. Below we briefly review the most important solution approaches for the VRPTW.

### Heuristic Approaches

Due to the complexity of the VRPTW, heuristic approaches were initially used in order to obtain feasible, but, in general, sub-optimal solutions. Heuristics still maintain an important role: (a) in providing good, feasible initial solutions to other methods (e.g. metaheuristics) or, (b) as embedded solution mechanisms in hybrid approaches and metaheuristics that provide quick solutions or local improvements. There are four significant classes of heuristics for the VRPTW:

1. Route construction heuristics start from an empty set of vehicles and a set of unrouted customers and iteratively combine routes with customers. Usually they are based on greedy procedures, in which the next best move is selected and implemented (Solomon, 1986; Solomon, 1987; Potvin and Rousseau, 1993).

2. Route improvement heuristics start from an initial feasible solution and operate on neighborhood solutions, i.e. solutions that are obtained by performing a single “move” (i.e. node swapping, arc interchange, etc). The first related references are those of Russell (1977), Cook and Russell (1978), and Baker and Schaffer (1986). Subsequently, several authors developed a wide range of route improvement heuristics, including Solomon *et al.* (1988), Savelsbergh (1985, 1990, 1992), Kindervater and Savelsbergh (1997), Cordone and Calvo (1996), and Thompson and Psaraftis (1993).
3. Two – phase Heuristics: They are usually classified in two groups: (a) Cluster-first, route-second, and (b) Route-first, cluster-second methods. In the former, the customers are first grouped into clusters taking into consideration resource capabilities and limitations; subsequently, a specific vehicle is assigned and routed for each cluster. In the latter case, initially a single, large route is constructed (using other heuristic approaches) that includes all customers and, then, it is divided into feasible vehicle routes.
4. Composite heuristics are combinations of route construction and improvement methods (e.g., Kontoravdis and Bard, 1995; Russell, 1995; Cordone and Calvo, 1997).

### **Metaheuristics**

Many authors have worked on avoiding the drawbacks of heuristic algorithms (e.g. reaching local minima) by adding further intelligence. Metaheuristics may overcome local minima by, for example, operating on large solution neighborhoods, exploring infeasible solutions, “travelling” to solutions stochastically, etc. In general, metaheuristics can be classified in three main classes (Toth and Vigo, 2002): (a) local search, i.e. simulated annealing (Chiang and Russell, 1996; Tan *et al.*, 2001), tabu search (Taillard *et al.*, 1997; Tan *et al.*, 2001), (b) population search, i.e. genetic search (Mester *et al.*, 2007; Homberger and Gehring, 2005), and (c) learning mechanisms, i.e. ant colony systems (Gambardella *et al.*, 1999). A comprehensive literature review on metaheuristics developed for the VRPTW is provided in Cordeau *et al.* (2002) and Toth and Vigo (2002), and has been further extended and updated by Gendreau *et al.* (2008).

### **Exact Approaches**

Exact approaches are based on network optimization and linear/integer/mixed programming. There are three main research directions for exact approaches (as stated in Larsen, 2001): Dynamic programming, Lagrangian relaxation, and column generation. Dynamic programming has been used by Kolen *et al.* (1987) to solve problems of up to 15 customers.

The other two methods are based on the decomposition principle, i.e. the main problem is decomposed into two (or more) distinct problems exploiting the special structure of VRPTW. The interested reader could find more information in Huisman *et al.* (2005).

In Lagrangian relaxation (Geofrion, 1974; Fisher, 1985) selected constraints are relaxed. That is, these constraints are removed from the constraint set and are converted to terms of the objective function, each multiplied by a penalty factor (the corresponding Lagrangian multiplier  $\lambda$ ). In this case, the master problem consists of finding the values of the Lagrangian multipliers (as well as the objective solution). In the VRPTW case, the subproblem is a shortest path problem with resource constraints (selected among the remaining constraints and related to route feasibility). Lagrangian relaxation has been addressed by Kohl (1995) and Kohl and Madsen (1997).

In column generation, the VRPTW is formulated through two dependent problem structures: (a) The Master Problem (MP), which is usually formulated as a set partitioning or set covering problem, and b) the sub-problem, which is a shortest path problem with time windows and capacity constraints (SPPTWCC). The use of column generation in VRPTW is overviewed in Kallehauge *et al.* (2005). The two problems interact by iteratively passing solutions to each other until the optimum is reached. Specifically, the SPPTWCC operates on a modified cost matrix, which is based on the real costs combined with the dual prices obtained from the MP. In turn, the MP incorporates the new feasible and negative-cost columns (routes) generated by the SPPTWCC, and is re-solved. This procedure is repeated in an iterative manner until no more feasible negative-cost columns are generated, and the optimum linear bound is found. To obtain integer optimal solutions, the entire procedure is embedded in a branch and bound scheme. A detailed description of the method is given in Section 2.2.1.

### **2.1.2 PERIODIC ROUTING PROBLEMS**

In addition to single-period (single day) routing problems, increased attention has been given to problems dealing with routing environments that incorporate several periods/days. The most known classes of periodic problems in the literature include: (a) The Inventory Routing Problem (IRP) (Dror *et al.*, 1985; Campbell and Savelsbergh, 2004), and (b) The Periodic Vehicle Routing Problem (PVRP) (Newman *et al.*, 2005; Christofides and Beasley, 1984). In these settings, the cost function concerns the overall horizon (several periods/days), and decisions to be made include both the assignment of customers to certain periods and the

routing of the customers within each period. In both the IRP and the PVRP, the frequency and timing of visiting a customer has major impact on the total routing cost.

### **The Inventory Routing Problem**

IRP combines inventory management with vehicle routing problems in a multi-period environment. Products are consumed by the customers at certain consumption rates. Each customer's storage capacity cannot be exceeded, and the fleet should serve all customers efficiently without allowing stock outs. The objective of the problem is to create the minimum-cost routes over the planning horizon, in order to replenish the customers efficiently. Based on this, many different variations can be formulated by incorporating different practical aspects such as inventory costs, stock out penalties, etc. Extended reviews can be found in Campbell *et al.* (1998) and Nori (1999).

The recent work of Bertazzi *et al.* (2008) overviews "simple" Inventory Routing Problems that involve a single product, a finite planning horizon, and deterministic consumption rates. Campbell and Savelsbergh (2004) present a two phase solution approach, in which customers to be served are selected first, and then routes are being generated for each period. Jaillet *et al.* (2002) address the IRP with satellite replenishment facilities, using a rolling horizon approach to estimate the total expected annual cost. The customers have a variable consumption rate and the cost of serving a customer is fixed (see Bard *et al.*, 1998). Using a rolling horizon framework, the authors estimate the cost for visiting customers in a repeated manner in order to minimize the annual delivery cost. Bertazzi *et al.* (2005) test different customer delivery policies as well as two different ways of decomposing the problem in order to minimize the total delivery cost over the time horizon. Lau *et al.* (2000; 2002) deal with time windows; in this case, the problem is decomposed into two sub-problems; the first defines the quantities to be delivered to customers with respect to inventory related costs, while the second constructs the routes for those customers selected by the first subproblem.

The presence of inventory and consumption considerations differentiates the IRP from the topic of this dissertation. Additional basic differences include:

- In IRP each customer is served more than once based on the rate of consumption, available capacity and routing aspects; in the problems discussed in this dissertation each customer is visited exactly once
- IN IRP customers are known *a priori*; in our (dynamic) case only a limited number of the customers are known (i.e. customers appear dynamically over time).

### The Period Vehicle Routing Problem

PVRP is a variation of VRP. The additional characteristics of PVRP are:

- The existence of a planning horizon  $P$ , i.e. a set of  $|P|$  consecutive periods (days)
- The existence of a set of alternative schedules ( $S$ ), i.e. combinations of days, during which a customer may be served.
- The frequency of service ( $f_i$ ) per customer  $i$ ; that is, customers may require to be served more than once within the planning horizon.

The above characteristics are discussed in Francis *et al.* (2008): The customers in PVRP are served multiple times over a certain time horizon (several days). Each schedule of the set  $S$  of possible schedules is defined by a vector of  $|P|$  elements, where each element  $a_{sp}$  is defined as:

$$a_{sp} = \begin{cases} 1 & \text{If period } p \text{ belongs to schedule } s \\ 0 & \text{Otherwise} \end{cases} \quad (2.11)$$

Note that when selecting a schedule  $s$  for a customer, then this customer will be visited in each  $p$  for which  $a_{sp} = 1$ . Based on the frequency of service ( $f_i$ ), not all schedules are compatible with customer  $i$ . That is, every customer  $i$  is compatible with a subset of schedules,  $S_i \subseteq S$  which satisfy the customer's frequency specifications. PVRP was introduced by Beltrami and Bodin (1974) and most recent algorithmic advance has been achieved by Baldacci *et al.* (2011). Several extensions have been proposed since, most of which embody additional operational or practical issues, such as multiple depots (Cordeau *et al.*, 1997; Hadjiconstantinou and Baldacci, 1998), and intermediate facilities for replenishment (Angelelli and Speranza, 2002). An interesting variation of the problem (PVRP with Service Choice) was introduced by Francis *et al.* (2006), in which the frequency of visits is not fixed but it is constrained to be no less than a lower service level limit; however visits may be performed in excess of this limit (i.e. when providing better service).

Several solution methods have been proposed for the PVRP based on heuristics, metaheuristics and mathematical programming techniques. Figure 2.1, from Francis *et al.* (2008), overviews the evolution of PVRP models and solution methods.

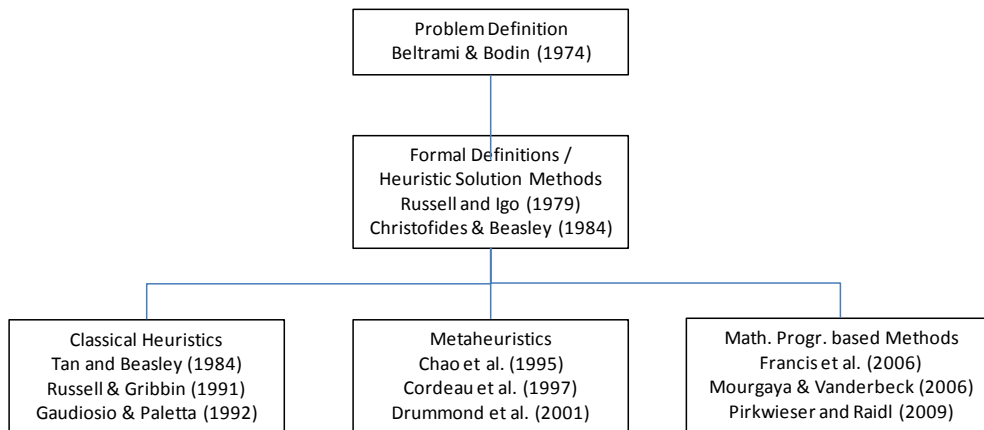


Figure 2.1: Evolution of models and solution methods for the PVRP (Francis *et al.*, 2008)

For example, column generation has been used both in exact and heuristic approaches. Pirkwieser and Raidl (2009) deal with the PVRP with time windows and provide the first exact approach using column generation. In this implementation only the linear bound is provided. Mourgaya and Vanderbeck (2007) use a column generation heuristic in order to solve the PVRP. Their approach consists of three stages, where initially customers are allocated to periods (days) to be serviced, then customers are assigned to the available vehicles, and, finally, the routing sequence of each vehicle is optimized. The first two stages consist of the tactical planning, and is the focus of the paper in order to minimize Euclidean distances among the customers assigned to each vehicle (in favor of geographical clustering) and to maintain a balanced workload among the vehicles. The third stage is not considered by the authors. In this respect, customer sequencing is not an objective of the problem.

Although there are many similarities between PVRP and the class of problems studied in this dissertation, including a) the assignment of customers to periods, b) the simultaneous solution of multiple periods, and c) considering the cost over the entire multi-period horizon, there are also distinct differences:

- In PVRP the customers can be visited more than once, while in our case each customer is visited exactly once
- In PVRP the customers are scheduled based on a predefined frequency (service patterns), while in our case each customer may be assigned to a certain set of consecutive periods (period window)
- In PVRP all customers are known *a priori* in contrast to our (dynamic) case where only a limited number of customers are known (i.e. customers appear dynamically over time).

## Other Related Problems

Other problems in the literature that consider multi-period environments include those addressing the combination of production and distribution. Gunnarsson and Rönnqvist (2008) introduced and solved a planning and distribution problem in a rolling horizon framework; Ravichandran (2007) proposes an ordering policy to optimize the expected operating profits through several periods (weeks). The problem is solved through multi-period dynamic programming. Zäpfel and Bögl (2008) address and solve an integrated vehicle routing and crew scheduling problem over a weekly horizon taking under consideration resource constraints (both vehicle- and personnel-related).

### 2.1.3 MULTI-PERIOD ROUTING PROBLEMS

In this Section we focus on problems of the literature that are highly related to this dissertation. We call these problems "multi-period routing problems" and we distinguish them from those overviewed in the previous Section.

Many researchers consider multi-period routing problems in which the customers become known progressively, and they approach them in a rolling horizon framework. In Figure 2.2, we classify multi-period vehicle routing problems in two different categories:

- The first category contains the MPVRP, in which all customers in the planning horizon of the next  $P$  periods are known. This problem can be considered as a special case of the PVRP.
- The second category is the Dynamic MPVRP, in which new customer requests arrive in each period of the planning horizon. Thus, when approaching this problem by solving an MPVRP for  $P$  periods in a rolling horizon framework, not all customers to be routed within these  $P$  periods are known. For a formal description of both cases, see Chapter 6.

Note that in the literature, another dynamic multi-period case is reported which regards the dynamic arrival of customers combined with the capability of modifying the schedule while vehicles are en-route (Angelesli *et al.*, 2009; Wen *et al.*, 2009). This case is highly related to the DVRP (Dynamic Vehicle Routing Problem); newly arrived customers can be either served by the vehicles that are en route (current schedule) or can be postponed for next periods based on their period window flexibility.

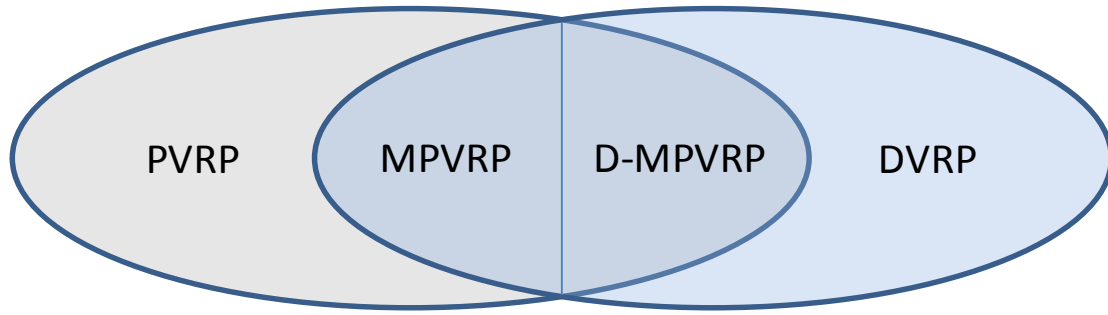


Figure 2.2: Related problems with the multi period routing problem

The existing literature in the field of multi-period routing problems is limited, as also stated by Bostel *et al.* (2008) and Wen *et al.* (2009). This limited literature refers to both single and multiple vehicle cases. Table 2.1 summarizes the main characteristics of the existing references on the MPVRP.

Table 2.1: Multi-Period Vehicle Routing Problems

	Multiple Vehicles	Time Windows	Period Window (# of periods)	Solution Procedure	Fixed Routes
Teng <i>et al.</i> (2006)			1 to 3	Heuristic/ Column Generation	
Angelelli <i>et al.</i> (2007)			1 or 2	Heuristic	
Andreatta and Lulli (2008)			1 or 2	Markov Process	
Tricoire (2006; 2007), Bostel <i>et al.</i> (2008)	✓	✓	1 or 2	Metaheuristic/Column Generation	
Wen <i>et al.</i> (2010)	✓		1 to 15 (avg. 2.5)	Heuristic	
Angelelli <i>et al.</i> (2009)	✓		1 or 2	Heuristic	
Athanasopoulos and Minis (2010)	✓	✓	5	Heuristic	✓

Teng *et al.* (2006) solved a single-vehicle multi-period routing problem that is based on the travelling salesman subset-tour problem (Mittenhal and Noon, 1992). In this case, customers are served once within certain predefined time periods. An additional profit is associated if service occurs within the predefined time periods. A column generation procedure is proposed and its efficiency is compared against heuristic methods.

Angelelli *et al.* (2007) considered a single-vehicle multi-period problem, and minimized the cumulative distance (cost). In their setting, customer requests arrive at the beginning of each period, and can be served in the next two consecutive periods. Different strategies for the allocation of customers (i.e. as-soon-as-possible, as-late-as-possible, and more sophisticated combinations) have been proposed and evaluated.

Andreatta and Lulli (2008) considered a special case of the multi-period routing problem with stochastic demand. Service is provided by a single vehicle either the day following the arrival

of the request (urgent service) or the subsequent day (regular service). The problem was formulated and solved by an aggregate Markov model using rewards.

Tricoire (2006; 2007) and Bostel *et al.* (2008) have studied another case of the MPVRP in which a certain number of customer requests are served over a horizon of  $P$  periods. Each request is served within a period window and a certain time window. A limited number of uncapacitated vehicles, starting from different locations (depots), are available for service. All vehicles have to respect certain labor rules, such as break intervals and maximum shift time. The problem is addressed both by a metaheuristic (memetic) algorithm and by an exact column generation approach. In the latter, the problem is divided in smaller problems (considering the next  $P$  periods) and is solved iteratively in a rolling horizon framework. Although this problem presents many similarities with our case, there are also many differences in the way the problem is approached and analyzed. Additionally, the current work addresses the generic case of the multi-period routing problem, based on which we investigate alternative column generation strategies for exploiting the special structure of multi-period problems.

Angelelli *et al.* (2009) expanded their research in MPVRP to address a dynamic setting with multiple vehicles, in which requests arrive as time unfolds and can be routed either in the period received or in the following one. A variable neighborhood search heuristic (see Mladenovic and Hansen, 1997) has been adapted with two different strategies: One that considers only the first period, and one that considers the first and the subsequent periods. Various objective functions have been proposed per strategy, which seek to maximize the number of customers as well as routing efficiency.

Wen *et al.* (2010) also address the MPVRP in a multiple vehicle setting. The problem was solved using the above variable neighborhood search heuristic along with a tabu search procedure (see Cordeau *et al.*, 1997). Alternative objective functions were addressed, including distance minimization, customer satisfaction and/or workload balancing.

Athanasopoulos and Minis (2010) solved a special case of the MPVRP, in which mandatory (inflexible) and flexible orders co-exist. The mandatory orders must be served within a certain period, and the flexible ones must be served within the entire planning horizon. This problem was addressed in two phases: Initially, predefined routes were constructed (to serve the mandatory customers) and, then, flexible customers were allocated in these routes at the appropriate periods of the planning horizon.

## **2.2 BRANCH AND PRICE THROUGH COLUMN GENERATION**

In the present Section we describe the Branch-and-Price method, an exact solution approach for vehicle routing problems, which is based on the column generation method combined with Branch-and-Bound. Section 2.2.1 reviews significant references for employing the column generation method to solve vehicle routing problems. Section 2.2.2 presents related references for the subproblem embedded within the column generation method (i.e. the shortest path problem with resource constraints). Finally, Section 2.2.3 presents the overall Branch-and-Price procedure that combines the aforementioned methods and provides integer optimal solutions to the routing problem. Note that these methods are further elaborated upon in Chapter 3 (column generation and shortest path problem with resource constraints), and Chapter 5 (Branch-and-Price).

### **2.2.1 COLUMN GENERATION**

A formal historical review of the column generation method is provided by Desrosiers and Lubbecke (2005). Ford and Fulkerson (1958) were the first to point out the advantages of decomposing the structure of linear programming problems. Later, Dantzig and Wolfe (1960) formalized their well-known decomposition scheme which, basically, generalized the column generation method. The first practical application using Column Generation was addressed by Gilmore and Gomory (1961; 1963) for the cutting stock problem. They provided a strategy to deal with a large linear program by splitting it into a master problem and several subproblems; in this scheme, information to the master problem is added in a repetitive manner by solving the subproblems. Luenberger (1989), Bradley *et al.* (1977) and Desaulniers *et al.* (2005) discuss the theoretical background of the decomposition and column generation methods. Additionally, Desaulniers *et al.* (2005) present research directions, as well as many applications of the column generation method.

Column generation is regarded as one of the most promising exact methods for addressing vehicle routing problems and, thus, it has attracted considerable attention by the related community. Desrosiers *et al.* (1984) and Agarwal *et al.* (1989) dealt with the VRPTW without capacity constraints, and the VRP, respectively. Desrochers *et al.* (1992) were the first to deal with the VRPTW. Since then a large number of references on the subject can be found (Kohl, 1995; Larsen, 2001; Feillet *et al.*, 2006; Chabrier, 2006). A survey on the applications of column generation for the VRPTW is provided in Kallehauge *et al.* (2005). A formal and analytical description of the column generation for the MPVRPTW is presented in Chapter 3.

### **2.2.2 SHORTEST PATH PROBLEM WITH TIME WINDOWS AND CAPACITY CONSTRAINTS (SPPTWCC)**

In vehicle routing problems, the SPPTWCC is typically the subproblem in the column generation framework, and, in general, it controls the efficiency of the method. The most common approaches to deal with the Elementary SPPTWCC are based either on Dijkstra (1959) algorithm (label setting), or on the Bellman-Ford algorithm (label correcting). While label correcting approaches create labels by processing nodes in an iterative manner, label setting algorithms select the next node to expand based on the lowest resource consumption. A label setting algorithm has been used by Larsen (2001) and Kohl (1995) in order to generate feasible routes (proposed ones) to be passed to the master problem. Note that in this case the master problem selects the best combination of routes among those proposed by the subproblem, in the context of column generation. Feillet *et al.* (2004) and Chabrier (2006) utilize a label correcting algorithm based on Desrochers *et al.* (1988; 1992) in order to generate proposed routes for the master problem. Both sets of authors have proposed modifications in order to include elementarity into the Desrochers *et al.* (1992) algorithm.

#### **Non-Elementary Shortest Path with Time Windows and Capacity Constraints (SPPTWCC)**

Due to the complexity of the ESPPTWCC, many researchers have studied the relaxed version of the problem, which results from eliminating the elementary constraints and, thus, allowing cyclic paths. Note that infinite cycling is anyway prohibited due to time and capacity constraints. This relaxation has been addressed by pseudo-polynomial solution algorithms (Desrochers *et al.*, 1992; Desrosiers *et al.*, 1995). Since then, a number of researchers used the SPPTWCC formulation to develop column generation algorithms for the VRPTW (Kohl, 1995; Larsen, 2001). In such formulations, the lower bound obtained was usually weaker than the lower bound obtained from elementary formulations. Thus, to improve the lower bounds, many researchers tried to deal with cycling using k-cycle elimination. 2-cycle elimination was initially developed by Houck *et al.* (1980) and Kolen *et al.* (1987). Their methods were embedded in the formulations of Desrochers *et al.* (1992), Kohl (1995), and Larsen (2001). Irnich (2001) and Irnich and Villeneuve (2003) developed the idea of k-cycle elimination. They also showed that by eliminating longer cycles, the lower bound obtained by the column generation process was drastically improved. k-cycle elimination is further discussed in Ziegelmann (2001), and Irnich and Desaulniers (2005).

**Elementary Shortest Path with Time Windows and Capacity Constraints (ESPTWCC)**

The elementary version of the SPPTWCC problem does not allow cycling, and, therefore, (a) no closed loops can be generated, and (b) no customers can be served more than once within the same trip. The problem has been proven to be NP-hard (Dror, 1994; Kohl, 1995) due to existence of the negative arc costs (see below) and the resource consumption property. The problem was initially solved by Beasley and Christofides (1989).

Based on the efficiency of the results obtained in the k-cycle elimination studies, Chabrier *et al.* (2002), Chabrier (2006), Feillet *et al.* (2004) and Feillet *et al.* (2005) have studied the elementary version of the SPPTWCC in order to obtain better lower bounds. Since computational complexity prevents the solution of large-scale problems (large number of customers, wide time windows, etc), attention has been paid to the development of intelligent dominance criteria, which reduce the search space without sacrificing optimality.

Boland *et al.* (2006) and Righini and Salani (2005) merged the non-elementary with the elementary SPPTWCC in order to utilize the strong features of each method; they proposed to solve the SPPTWCC and add elementary constraints to certain customer nodes as the solution procedure progresses. Some of the most promising approaches to-date for the VRPTW have been provided using the partial elementarity principle (Desaulniers *et al.*, 2006) or full elementarity (Feillet, 2005).

*Related references to our multi-period setting include the following:* In Pirkwieser and Raidl (2009) new columns, which are generated by each subproblem, are transferred to all other subproblems without solving each subproblem separately. Note that the authors do not elaborate more on the strategy and whether or not some columns may be eliminated for some periods due to infeasibilities in schedule and/or visit frequency. Mourgaya and Vanderbeck (2007) propose a "cyclic generation strategy", in which one subproblem is solved and the generated columns are provided to the other subproblems. Note that in both references, the subproblems to be solved generate feasible columns to all periods, which is not the case in our problem.

**2.2.3 BRANCH AND PRICE**

When solving the relaxed Multi-Period Routing Problem with Resource Constraints (e.g. time windows and capacity constraints), the methods described in Chapter 3 and 4 provide a lower bound (LB) of the solution. In case this LB corresponds to an integer solution, the optimal

solution to the problem has been found and no further exploration of the solution space is necessary. In case the LB corresponds to a fractional solution, then it is a bound of the minimum cost of the optimal solution, and further investigation of the solution space is necessary to obtain the integer optimum.

Branch and Bound (B&B) is a simple generic algorithm that explores the solution space in order to provide integer solutions. In the problems related to this dissertation, B&B guarantees that the optimal integer solution will be obtained if all the columns (routes) are known. In the case of Column Generation (CG), in which only a small portion of the total feasible columns (routes) is available, B&B cannot guarantee optimality. A straightforward implementation of the B&B procedure on the final optimal (lower bound) solution does not guarantee that the optimal integer solution will be found, since only a subset of the feasible routes (columns) has been generated by the subproblem. Thus, in order to be able to obtain the optimal integer solution, the column generation method should be applied to each node of the B&B tree, allowing new columns (routes) to be created. This procedure is called Branch and Price (Barnhart *et al.*, 1998; Desrosiers *et al.*, 1995; Soumis, 1997; Danna and Le Pape, 2005) and was initially proposed by Desrosiers *et al.* (1984) for the VRPTW. In addition to the exact B&P approach many researchers have proposed mixed approaches in order to speed up the solution procedure. Danna and Le Pape (2005) proposed a mixed scheme, which uses B&P, an MIP heuristic solver and a metaheuristic, while Jepsen *et al.* (2008) used the Chvatal rank-1 cuts (subset row inequalities) in order to obtain better lower bounds through a branch-and-cut-and-price algorithm. A more detailed description of the technical background of B&P is provided in Chapter 5.

## 2.3 RESEARCH CONTRIBUTION

Our contribution to the study of multi-period routing problems provides insights in the mathematical formulation, the exact column generation approach, the multi-period related acceleration techniques, the theoretical understanding of the multi period setting, as well as into significant practical aspects of the problem (limited resources, predefined routes). Although elements of this problem have been investigated in the literature, the dissertation addresses new aspects of the problem and makes the following contributions:

1. We propose a decomposed mathematical formulation for the generic case of the multi-period routing problem. This formulation generalizes the one presented in Bostel *et al.*

(2008), and, as such, can be used as the basis for describing several variations of multi-period routing problems.

2. By understanding and taking advantage of the special structure of the multi-period problem, i.e. (a) the independence of the column generation subproblems per period and (b) the flexibility of customers to be routed in several alternative periods, we developed two different strategies to solve the subproblems, the most "costly" part of the column generation method. These strategies are: (i) one subproblem is solved, and the generated columns are transferred to the others; (ii) a single, unified subproblem is solved, considering all periods simultaneously. All required modifications to the classical column generation approach for solving multi-period problems are studied and discussed.
3. We compare these two strategies with two existing approaches, whereby each subproblem is solved separately and sequentially, or in parallel. Specific instances, which address different patterns of period flexibility, were created in order to benchmark the aforementioned strategies. We show that significant computational savings can be achieved by using the proposed strategies, especially in cases in which customers have increased period flexibility. These savings are relevant to both determining the lower bound, and the optimal integer solution. This result has been obtained through extensive experimental investigation that uses Solomon benchmarks (and their extended versions), in order to generate appropriate test cases for large numbers of consecutive periods and different customer period window patterns.
4. Additionally, a simple pruning heuristic is proposed in order to accelerate the solution procedure. The performance of this heuristic, with respect to the efficiency of both the final solution and the required computational time, indicates its suitability for solving multi-period vehicle routing problems in long horizons using a rolling horizon framework.
5. As far as the rolling horizon framework is concerned, we focused on two arrival patterns of customer requests: (a) the quasi-static (MPVRP), and (b) the Dynamic MPVRP. For the first case, we propose and discuss three theoretical statements concerning the implementation horizon ( $M$ ) and the planning horizon ( $P$ ), which are key parameters in the rolling horizon implementation. These statements establish the principles of applying the proposed methods to solve routing problems in long-term time horizons.
6. In order to address significant practical aspects, we modify the MPVRPTW model to take into consideration cases in which not all customer orders can be served within the planning horizon (e.g., the number of the consecutive periods that are simultaneously considered in each MPVRPTW). This new modification is achieved by introducing

appropriate penalty functions for the unserved customers taking into account their period flexibility. Five such penalty functions are proposed, analyzed and compared.

7. Significant experimental results were obtained considering both cases: (a) the quasi-static (MPVRP), and (b) the Dynamic MPVRP. For these cases, larger planning horizons result in lower routing costs, validating the appropriateness and the efficiency of the proposed methods. The experimental results follow the same pattern for different time windows, as well as for different geographical distribution of customer orders. The above conclusions are also validated through appropriate statistical analysis.
8. We also apply our multi-period approach to the case of the multi-period routing problem with pre-assigned customers, which has significant practical applications. A new solution approach is provided, based on a modified version of the column generation procedure proposed for the general MPVRPTW. Extensive experimental investigation indicates that significant cost savings can be achieved by considering wider planning horizons in the planning process.

## **Chapter 3:      THE MULTI-PERIOD VEHICLE ROUTING PROBLEM WITH TIME WINDOWS**

The problem addressed here is related to environments in which a fleet of vehicles serves a set of customers over a multiple-period (planning) horizon. In this environment, every assignment has some time-related flexibility, i.e. it can be assigned/routed to one or more periods within the planning horizon. The challenge that arises is to simultaneously assign customers to periods (days) and route the vehicles to serve these customers. The objective of the problem is to minimize the total cumulative routing cost throughout the planning horizon.

The described problem is related to the well-known VRP and PVRP cases. However, VRP requires that each customer is served within a single period, while PVRP has a predefined periodic visit pattern for every customer. The problem has been formulated here as the general Multi-Period Vehicle Routing Problem with Time Windows (MPVRPTW). Versions of the MPVRP with or without time windows have been addressed in the literature by Angelelli *et al.* (2007), Tricoire (2006) and Wen *et al.* (2010). In this dissertation we present alternative exact solution methods that use column generation and exploit the special multi period structure of the problem (see Chapter 4).

This chapter has dual purpose, that is: (a) To define the general mathematical model of the MPVRPTW and (b) to present the basic column generation solution method that forms the foundation of the novel alternative approaches discussed in Chapter 4. In the remainder of this Chapter, Section 3.1 presents the mathematical model of MPVRPTW and Section 3.2 describes the basic column generation method for solving the MPVRPTW. Section 3.3 presents the solution mechanism for the Elementary Shortest Path Problem with Time Windows and Resource Constraints (ESPPTWCC), which is the core component of the column generation framework in our case. Finally, Section 3.4 presents a conceptual synthesis of the overall column generation method for the MPVRPTW.

### **3.1 MATHEMATICAL FORMULATION**

Our formulation for the MPVRPTW is based upon the formulation of the VRPTW presented by Cordeau *et al.* (2002) and by Wen *et al.* (2010) for the MPVRP. Cordeau *et al.* (2002)

model is further expanded to accommodate: (a) the multiple periods, and (b) the period window constraints of the customers.

Consider a planning horizon of  $|P|$  periods and let  $p_c$  be the current period. Also assume that all customers should be served over the next  $P$  periods, that is in the planning horizon  $[p_c + 1, p_c + P]$ . For simplicity, and without loss of generality, the current period (planning period) is set to  $p_c = 0$  and, the planning horizon is simplified to  $[1, P]$ . The basic notation of the problem is given below.

$H$	Set of $P$ consecutive periods (planning horizon)
$N = \{1, \dots, n\}$	Available customers (orders) at the beginning of period 1
$W = N \cup \{0, n + 1\}$	Set of vertices, including the starting and ending depot. Every vehicle route starts from the starting depot (node 0) and finishes at the ending depot (node $n + 1$ ). Two nodes are used for the depot in order to allow a vehicle to remain inactive within a period, i.e. establish a $(0, n + 1)$ connection (see Cordeau <i>et al.</i> , 2002)
$A = \{(i, j) : i, j \in W\}$	Set of arcs connecting all vertices (nodes) in $W$
$I_i = [\xi_i^s, \xi_i^e]$	Period window of customer $i$ ; where $1 \leq \xi_i^s \leq \xi_i^e \leq P$ . Note that when a customer requests service within $[\xi_i^s, \xi_i^{e'}]$ , where $\xi_i^{e'} > P$ , this customer's period window will be reduced to $[\xi_i^s, P]$ <sup>1</sup>
$c_{ij}$	Cost for traversing arc $(i, j), \{i, j \in W\}$
$t_{ij}$	Time for traversing arc $(i, j), \{i, j \in W\}$ including the service time of customer $i$ . For nodes 0 and $n + 1$ the service time equals zero.
$d_i$	Demand of customer $i, \{i \in N\}$
$K_p$	Set of $ K_p $ available vehicles per each period $p, \{p \in H\}$
$Q_k^p$	Capacity of vehicle $k$ during period $p$
$[a_i, b_i]$	Time window of customer $i$ , same for each period within $I_i$ ; for nodes 0 and $n + 1$ , $a_0 = a_{n+1}$ is the earliest time each vehicle can leave the

---

<sup>1</sup> In Chapter 6 the more generic case, where  $\xi_i^{e'} > P$ , is discussed along with alternative solution procedures.

depot, and  $b_0 = b_{n+1}$  is the latest time each vehicle can return to the depot.

Two sets of variables are used for the model:  $x_{ijpk}$  equals one if route  $k$  of period  $p$  traverses arc  $(i, j)$  and zero otherwise;  $s_{ipk}$  represents the start of service for customer (node)  $i$  by vehicle  $k$  within period  $p$ .  $s_{ipk}$  is set to zero if node  $i$  is not served by a vehicle (say  $k$ ) within a period (say  $p$ ).

The problem's objective is to minimize the total cumulative routing cost over the planning horizon, and is given by:

$$\min(z) = \sum_{p \in H} \sum_{k \in K_p} \sum_{(i,j) \in A} c_{ij} x_{ijpk} \quad (3.1)$$

### Constraints

$$\sum_{p \in I_i} \sum_{k \in K_p} \sum_{j \in NU\{n+1\}} x_{ijpk} = 1 \quad \forall i \in N \quad (3.2)$$

$$\sum_{p \notin I_i} \sum_{k \in K_p} \sum_{j \in NU\{n+1\}} x_{ijpk} = 0 \quad \forall i \in N \quad (3.3)$$

$$\sum_{j \in NU\{n+1\}} x_{0jpk} = 1 \quad \forall p \in H, \forall k \in K_p \quad (3.4)$$

$$\sum_{i \in NU\{0\}} x_{ijpk} - \sum_{i' \in NU\{n+1\}} x_{ji'pk} = 0 \quad \forall p \in H, \forall k \in K_p, \forall j \in N \quad (3.5)$$

$$\sum_{j \in NU\{0\}} x_{j,n+1,pk} = 1 \quad \forall p \in H, \forall k \in K_p \quad (3.6)$$

$$\sum_{i \in N} d_i \sum_{j \in NU\{n+1\}} x_{ijpk} \leq Q_k^p \quad \forall p \in H, \forall k \in K_p \quad (3.7)$$

$$s_{ipk} + t_{ij} - M(1 - x_{ijpk}) \leq s_{jpk} \quad \forall p \in H, \forall k \in K_p, \forall (i, j) \in A \quad (3.8)$$

$$a_i \sum_{j \in NU\{n+1\}} x_{ijpk} \leq s_{ipk} \leq b_i \sum_{j \in NU\{n+1\}} x_{ijpk} \quad \forall p \in H, \forall k \in K_p, \forall i \in N \quad (3.9)$$

$$a_i \leq s_{ipk} \leq b_i \quad \forall p \in H, \forall k \in K_p, i \in \{0, n+1\} \quad (3.10)$$

$$x_{ijpk} \in \{0, 1\} \quad \forall p \in H, \forall k \in K_p, (i, j) \in A \quad (3.11)$$

Objective function (3.1) expresses the total routing cost over the entire planning horizon. Constraints (3.2) and (3.3) specify that each customer will be visited once (one route and during one period only) within the corresponding period window for that customer.

Constraints (3.4) and (3.6) specify that each vehicle departs from the starting depot (0) and ends at the ending depot ( $n + 1$ ). Constraints (3.5) are the flow conservation constraints for every route. Constraint (3.7) secures that the capacity of each vehicle is not violated. Note that subtour elimination constraints are not needed in this model, due to the time windows and the variables  $s_{ipk}$ . Constraints (3.8), (3.9) and (3.10) ensure that each customer is served within its time window. Note that  $M$  represents a large number, which should be larger than  $M_{ij} = \max(b_i + t_{ij} - a_j, 0)$  for each arc  $(i, j)$ . Finally, Constraints (3.11) force the flow variables to assume the binary values  $\{0, 1\}$ .

The solution procedure that is described in the present dissertation is based on the Column Generation (CG) method which decomposes the linear relaxation of the above problem to a Master Problem (MP) and several subproblems (SP). Since MP integrality constraints have been relaxed, the latter is solvable using linear programming techniques; integer solutions are provided through a Branch and Price framework based on the lower bound provided by the CG method (See Chapter 5). Also note that, although the decomposition of problem (3.1)-(3.12) into a Master Problem and the relevant subproblems, is fundamentally based on the Dantzig-Wolfe decomposition (Dantzig and Wolfe, 1960), a more straightforward approach will be used following the approach of Desrochers *et al.* (1992) and Desaulniers *et al.* (2005).

## **3.2 LOWER BOUNDS THROUGH COLUMN GENERATION**

This section discusses the essential features of the MPVRPTW within a column generation framework. This decomposition is inspired by related approaches proposed by some authors to address the VRPTW (Desrochers *et al.*, 1992; Larsen, 2001; Feillet *et al.*, 2005, Chabrier, 2006), the PVRP (Mourgaya and Vanderbeck, 2007; Pirkwieser and Raidl, 2009) and the MPVRPTW (Tricoire, 2006; Tricoire, 2007; Bostel *et al.*, 2008).

### **3.2.1 THE PROPOSED MASTER PROBLEM**

In the decomposed model, the Master Problem (MP) includes linking Constraints (3.2) and (3.8), which cannot be treated independently by the subproblems. The reason is that these constraints cannot be separated either per period  $p \in P$  or per vehicle  $k \in K_p$ , since each is expressed as a sum over all periods and vehicles. It is also noted that in the model of Eqs (3.1)-(3.12) the vehicle constraints are not defined explicitly as a separate set of constraints but are rather implied by the existence of the set  $K_p$  for each period. That is, the objective

function, as well as the constraints, are defined for a limited set of vehicles per period  $p$ , of size  $|K_p|$ .

The MP is a Set Partitioning Problem, since every customer should be serviced exactly once. Desrochers *et al.* (1992) recommended the Set Covering formulation for the VRPTW, which allows visiting each customer more than once. This modification was later adopted by many researchers (Feillet *et al.*, 2005) and allows for simpler initial solutions to the linear relaxation of problem. In addition, as stated in Feillet *et al.* (2005) the set covering formulation is also preferable with respect to convergence, since the shadow prices related to Constraint (3.16) are always non-negative, which is not true in the Set Partitioning formulation. The latter property stabilizes the solution process (i.e. providing for smoother convergence of the shadow prices to their optimal values), thus leading to more efficient computations of the shadow prices.

Note also that the Set Covering formulation Problem is optimal with respect to the Set Partitioning Problem. Although Set Covering allows customers to be assigned to more than one vehicle, it is straightforward to show that in the optimal solution every customer will participate only once. For example, consider a solution that includes two routes that contain a common customer. There is always a better solution (with lower routing cost) in which the common customer has been eliminated from one of the former two routes. This property holds when the triangular inequality holds. Based on this fact, even if the linear programming solution procedure travels through solutions with multiple visits to customers, finally it will conclude to a solution that visits each customer just once (Feillet *et al.*, 2005).

For the reasons stated above, in this work we adopt the Set Covering formulation.

Let  $p = 1, \dots, P$  denote the periods of the planning horizon and  $\Omega_p$  the set of all feasible routes for period  $p$ . Coefficients  $a_{ir}^p$  are defined as:

$$a_{ir}^p = \begin{cases} 1 & \text{if customer } i \text{ is included in route } r \text{ in period } p \\ 0 & \text{otherwise} \end{cases} \quad (3.12)$$

Variables  $x_r^p$  are defined as:

$$x_r^p = \begin{cases} 1 & \text{if route } r \text{ of period } p \text{ is used in the solution} \\ 0 & \text{otherwise} \end{cases} \quad (3.13)$$

If  $C_r^p$  denotes the cost of route  $r$  of period  $p$ , then the objective function of the Master Problem is of the following form:

$$\min \sum_{p=1}^P \sum_{r \in \Omega_p} c_r^p x_r^p \quad (3.14)$$

$$\sum_{r \in \Omega_p} x_r^p \leq K \quad \forall p \in P \quad (3.15)$$

$$\sum_{p=1}^P \sum_{r \in \Omega_p} a_{ir}^p x_r^p \geq 1 \quad \forall i \in N \quad (3.16)$$

$$x_r^p = \{0, 1\} \quad (3.17)$$

The objective function (3.14) minimizes the total cumulative routing cost. Constraints (3.15) restrict the number of vehicles to be used in each period to the available fleet, and Constraints (3.16) are the set covering constraints. Eliminating binary Constraints (3.17) relaxes the problem and permits it to be solved using well known linear programming techniques.

In the aforementioned formulation, we have assumed that all feasible routes for every period ( $\Omega_p$ ) are known *a priori*. Given, however, that the  $\Omega_p$  sets will not be known in their entirety, we denote as  $\Omega'_p$  a subset of  $\Omega_p$ . Every set  $\Omega'_p$  for every period  $p$  contains feasible routes for this period. Additionally, the initial collection of sets  $\Omega'_p$  should contain a feasible initial solution. Note that, in our case, due to the limitation on the number of vehicles (Constraints 3.16) the use of the trivial initial solution that comprises one route per customer (i.e., *depot* – *i* – *depot*) is not feasible. The problem involving the  $\Omega'_p$ , instead of the  $\Omega_p$ , sets is called the Restricted Master Problem (RMP).

In order to better explain the definition of the MP and its relationship to the RMP, an illustrative example is given in Fig. 3.1; the figure shows the form of the coefficient matrix of the linear model along with the Right Hand Side (RHS) vector. Note that each element of the first  $p$  elements ( $Vp$ ) of the RHS represent the number of available vehicles per period  $p$ . The first  $|P|$  rows represent the period constraints (3.15), while the next  $|N|$  rows represent the set-partitioning constraints. Thus, each column comprises a vector of  $|P| + |N|$  elements. The first  $|P|$  elements of every column are all zeros except from one element that is related to each period and is equal to one. For example, for period 3, the vector of the  $|P|$  elements is  $[0 \ 0 \ 1 \ 0 \dots 0]^T$ . The remaining  $|N|$  elements represent the actual route as described in (3.12). Dark areas are the known routes per period, forming the RMP constraint set. The latter contains only a limited number of the spectrum of feasible routes per period (blue plus grey areas, MP).

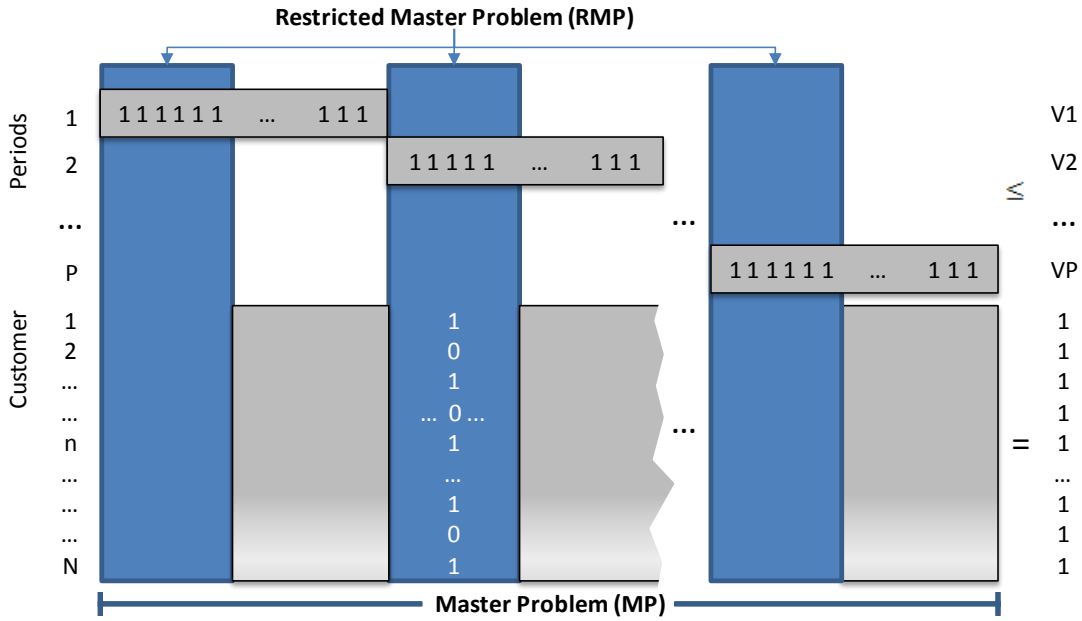


Figure 3.1: Master Problem (MP) and Restricted Master Problem (RMP). Set partitioning formulation.

In order to reach optimality for the MP by solving the RMP, we should be able to generate good quality columns that are not known (from the grey areas), include them to the current RMP (blue areas) and solve the new updated RMP. This could be done iteratively until the optimal solution is obtained. The role of the subproblems is precisely the generation of these new columns (routes per each period), as discussed in Section 3.2.2. This can be repeated until no new columns (routes) with negative reduced cost can be provided by the subproblems. Then the optimal lower bound has been obtained.

It is noted that solving the subproblem is equivalent to selecting a new basic variable in a classic simplex procedure. The termination criterion is also equivalent; that is, all reduced costs are non-negative or no new columns with negative shadow prices can be generated by the subproblems. This will be further explained in Section 3.2.2. Luckily, and as practice has shown, only a portion (hopefully restricted) of the total feasible routes of  $\Omega_p$ s will be generated prior to reaching optimality.

Bostel *et al.* (2008) and Tricoire (2007) have proposed a similar decomposed formulation for a special case of the multi-period routing problem with multiple depots. In their model a different constraint on the vehicle availability is proposed: Instead of defining a set of available vehicles per period, they define a (larger) set of "resource-days". Each "resource-day" represents a combination of periods and vehicles, i.e. the availability of one vehicle during one period. In their formulation, one subproblem is solved per each resource-day. That

is, given  $p$  periods and  $K$  vehicles (same number of vehicles per period) a total number of  $p \times K$  subproblems need to be solved. This increases significantly the complexity of the formulation and, indeed, of the solution algorithm in relation to the one proposed above.

### 3.2.2 THE SUBPROBLEMS

As already discussed, the MP and the RMP consider only the linking constraints. All the remaining problem constraints are transferred to the subproblem(s) and they form the Elementary Shortest Path Problem with Time Windows and Capacity Constraints (ESPPTWCC).

Note that the solution provided by the current RMP is optimal with respect, of course, to the columns (routes) that are contained in the  $\Omega'_p$  sets. In order to check if this solution is globally optimal for the MP, we should calculate the reduced costs ( $\bar{c}_r^p$ ) of each non-basic route  $\tilde{r} \in \Omega_p$  of each period  $p$  (note that  $\Omega_p$  contains also the routes that do not yet exist in the current RMP). The reduced cost for each column related to route  $r$  and period  $p$  is given by the following equation:

$$\bar{c}_r^p = C_r^p - \sum_{i \in N} a_{ir}^p \pi_i - \sigma_p \quad \forall r \in \Omega_p, \forall p \in P \quad (3.18)$$

where  $\pi_i$  and  $\sigma_p$  are the shadow (dual) prices related to customer Constraints (3.17) and period Constraints (3.16), respectively. The calculation of Eq. (3.18) for every route contained in the current RMP is straightforward, since all elements are known.

In the classical simplex procedure, having calculated all reduced costs, and in order to insert a route  $\tilde{r}$  in the basic solution (i.e. make a non-basic variable, basic), its reduced cost ( $\bar{c}_r^p$ ) should be negative. It is already known that the reduced cost  $\bar{c}_{r'}^p$  of each non-basic route  $r' \in \Omega'_p$  in every period  $p$  (routes in the blue area of Fig. 3.1) is non-negative, and, therefore, should not be considered for inclusion in the basis of the current RMP. Thus, routes  $\tilde{r} \notin \Omega'_p$  that have not yet been included in the RMP (grey areas of Fig. 3.1) should be generated, along with their reduced costs. To do so, for each period  $p$  a minimization problem (**subproblem**) is solved, in which the route  $\tilde{r}_p^*$  with the minimum reduced cost ( $\bar{c}_{\tilde{r}_p^*}^p$ ) is derived, that is:

$$\bar{c}_{\tilde{r}_p^*}^p = \min_{r \in \{\Omega_p \setminus \Omega'_p\}} (\bar{c}_r^p) \quad \forall p \in P \quad (3.19)$$

Then, the overall minimum reduced cost  $\bar{c}_{\tilde{r}^*}^p$ , over all periods is calculated:

$$\bar{c}_{\hat{r}^*}^p = \min_{p \in P}(\bar{c}_{\hat{r}^*}^p) \quad (3.20)$$

Note that thus far, this procedure resembles a typical iteration of the classical simplex procedure. The twist is that the routes  $\hat{r}$  (i.e. columns) are not known. Consider the following reformulation of the routing cost of route  $r$  in period  $p$  with respect to the arc cost coefficients:

$$C_r^p = \sum_{i \in N} a_{ir}^p c_{ij} \quad j \in i^+, \forall r \in \Omega_p, \forall p \in P \quad (3.21)$$

where  $j \in i^+$  denotes that customer  $j$  is the next customer to be visited after customer  $i$  in route  $r$  of period  $p$ , and  $c_{ij}$  is the cost traversing arc  $(i, j)$ . Combining Eq. (3.21) and Eq. (3.18) we obtain:

$$\bar{c}_r^p = \sum_{i \in N} a_{ir}^p c_{ij} - \sum_{i \in N} a_{ir}^p \pi_i - \sigma_p = \sum_{i \in N} a_{ir}^p (c_{ij} - \pi_i) - \sigma_p \quad (3.22)$$

Thus, Eq. (3.19) for the subproblem per period  $p$  becomes:

$$\bar{c}_{\hat{r}^*}^p = \min_{\hat{r}} \left( \sum_{i \in N} a_{i\hat{r}}^p c'_{ij} - \sigma_p \right) \quad (3.23)$$

Cost factors  $c'_{ij}$  are the *modified costs* coefficients of each arc  $(i, j)$ , which can be negative.

$$c'_{ij} = \begin{cases} c_{ij} - \pi_i & \forall i \in N \setminus \{0, n+1\}, j \neq 0 \\ c_{ij} & i = 0, j \neq 0 \\ +\infty & i = n+1 \\ +\infty & j = 0 \end{cases} \quad (3.24)$$

Note that the coefficients  $a_{i\hat{r}}^p$  in Eq. (3.23) are not known. The scope of each subproblem is to define the values of coefficients  $a_{i\hat{r}}^p$  that minimize the subproblem, i.e. the minimal (shortest) path, and the relevant reduced cost.

In order to further transform Eq. (3.23) into a double-index mathematical formulation, coefficients  $a_{i\hat{r}}^p$  are substituted by the arc variables  $x_{ij}$ . Consider for example, a route that visits customers 2, 4 and 6 from a set of six customers, that is  $r = [D \ 2 \ 4 \ 6 \ D]$ , where D represents the depot, and the route's related cost  $C_r$ . The equivalent representation of this route, in terms of the coefficients  $a_{i\hat{r}}^p$  is  $[0 \ 1 \ 0 \ 1 \ 0 \ 1]^T$ . Note that the depot is not represented in the  $a_{i\hat{r}}^p$  coefficients. This route is also defined by the arc variables  $x_{12}, x_{24}, x_{46}$  and  $x_{61}$  and the relevant route cost ( $C_r = c_{12} + c_{24} + c_{46} + c_{61}$ ). Note that the subscript ( $k$ ) for vehicle is

dropped for each subproblem, since the vehicles are identical and the relevant constraints remain in the RMP. Then, Eq. (3.23) takes the following form:

$$\min \sum_{i \in N_p \cup \{0\}} \sum_{j \in N_p \cup \{n+1\}} c'_{ij} x_{ijp} - \sigma_p \quad (3.25)$$

Note that Eq. (3.25) is solved per each period  $p$ . Initially and in order to respect period windows, in the subproblem of each period  $p$  -Eq. (3.25)- only the feasible customers (i.e. customers that are allowed to be routed in period  $p$ ) are included. The set of the feasible customers per each period is represented as  $N_p$ . Based on that, there is no need to include additional constraints for period window feasibility. Of course, a customer will participate in as many subproblems as the periods in its period window. Additionally, the cost, modified cost and time matrix consisting of coefficients  $c'_{ij}$ ,  $c_{ij}$  and  $t_{ij}$  are defined independently  $\forall p$  and contain only the links among the customer set  $N_p \cup \{0, n+1\}$ .

In the RMP only feasible routes are allowed to be inserted, and thus Eq. (3.25) should be restricted to generate only feasible routes. For that, Constraints (3.26) to (3.32) are added to each subproblem (see below). Note that these constraints secure the feasibility of a single path (route), are derived from the original problem formulation (3.1) - (3.12), and are defined for each period  $p \in P$ .

$$\sum_{j \in N} x_{0jp} = 1 \quad (3.26)$$

$$\sum_{i \in N_p \cup \{0\}} x_{ijp} - \sum_{i' \in N_p \cup \{n+1\}} x_{ji'p} = 0 \quad \forall j \in N_p \quad (3.27)$$

$$\sum_{i \in N} x_{i,n+1,p} = 1 \quad (3.28)$$

$$s_{ip} + t_{ij} - K(1 - x_{ijp}) \leq s_{jp} \quad \forall i, j \in N_p \cup \{0, n+1\} \quad (3.29)$$

$$a_i \leq s_{ip} \leq b_i \quad \forall i \in N_p \cup \{0, n+1\} \quad (3.30)$$

$$\sum_{i \in N_p} d_i \sum_{j \in N_p \cup \{0, n+1\}} x_{ijp} \leq Q \quad (3.31)$$

$$x_{ijp} \in \{0, 1\} \quad \forall i, j \in N_p \quad (3.32)$$

Also note that, without loss of generality, we can drop subscript ( $p$ ) from variables  $x_{ijp}$  since every subproblem regards a specific period  $p$ .

The objective function of (3.25) expresses the route with the minimum modified cost for period  $p$ . Constraints (3.26) and (3.28) specify that the route starts and ends at the depot.

Constraints (3.27) are the flow conservation constraints. Constraints (3.29) and (3.30) ensure that every customer will be served within its time window. Constraint (3.31) respects the capacity of the vehicle assigned to the route. Finally, Constraint (3.32) forces the flow variables to assume binary values  $\{0, 1\}$ .

#### Note

By observing the original problem formulation (3.1) to (3.12), subproblem independence can be defined both per period and per vehicle. For problems for which identical vehicles are considered, only one subproblem needs to be solved per period, as in the case of VRPTW (Kallehauge *et al.*, 2005, Larsen, 2001, Desrochers *et al.*, 1992), where only one period is considered. This is also the case in the current problem, in which the subproblem per period  $p$  is different only with respect to the customers available to be routed. The solution of each subproblem will provide candidate routes that can be assigned to every available vehicle in the related period.

### **3.3 THE ESPPTWCC – SOLUTION PROCEDURE**

The subproblem described above is an Elementary Shortest Path Problem with Time Windows and Capacity Constraints (ESPPTWCC). Thus, existing, efficient algorithms for the ESPPTWCC can be utilized. The method that has been implemented in the present research is based on the label correcting algorithm of Feillet *et al.* (2004). The column generation scheme, which comprises the RMP and the subproblems, has two significant characteristics.

- It is guaranteed to converge to the optimal solution (optimal lower bound)
- Since each subproblem is NP-hard, practical convergence of the column generation scheme to the optimal solution depends on the speed (efficiency) of solving the subproblems.

#### **Note on complexity**

Although the shortest path problem (SPP) is polynomial [e.g. it can be solved in  $O(nm)$  by the Bellman-Ford algorithm], the inclusion of the time-window and capacity constraints turns ESPPTWCC into an NP-Hard problem. By relaxing the elementarity constraints, i.e. SPPTWCC, the problem remains NP-hard, but can be solved in a pseudo-polynomial time. A detailed description of the complexity of the ESPPTWCC is provided in Larsen (2001).

Therefore, in practice, optimality depends strongly on the computational efficiency of solving ESPPTWCC. As a consequence, our effort in this part of the research has focused on ways to improve the efficiency of the algorithm to solve ESPPTWCC in order to be able to solve optimality problems of practical importance. To do so, we have enhanced the algorithm considerably by:

- Incorporating successful improvements from a wide spectrum of work from the literature (discussed below), and
- Developing novel improved column generation structures, based on the intrinsic characteristics of the multi-period problem in hand (see Chapter 4)

The label correcting algorithm of Feillet *et al.* (2004; 2005) adopted as the basis of the solution algorithm, along with several other improvements incorporated in our work, are described below.

### **Solution Procedure**

Each partial path ( $\delta$ ) ending at node  $i$  is associated with a label  $L_{\delta i} = [\bar{c}_{\delta i}, t_{\delta i}, d_{\delta i}]$ , representing the accumulated reduced cost, time, and demand between the origin and the last node ( $i$ ) of partial path  $\delta$ . Note that in (E)SPPTWCC, in contrast to the generic single-source, single-destination shortest path problem (Cormen *et al.*, 2003), each node is associated with more than one labels, due to the resource consumption (time, demand) limitations. The existence of multiple labels is the major reason for the high complexity of the problem. In order to reduce the number of these labels, we use dominance criteria.

Initially the method starts from label  $L_0 = [0, 0, 0]$ , corresponding to the origin node, and extends to all other graph nodes (except to node  $n + 1$ ). When extending label  $L_{\delta i} = [\bar{c}_{\delta i}, t_{\delta i}, d_{\delta i}]$ , to a node  $j$ , then the new label  $L_{\delta' j}$  representing partial path  $\delta'$  ending at node  $j$  is given by the following equations:

$$\bar{c}_{\delta' j} = \bar{c}_{\delta' i} + (c'_{ij}) \quad (3.33)$$

$$t_{\delta' j} = \max \{t_{\delta' i} + t_{ij}, a_j\} \quad (3.34)$$

$$d_{\delta' j} = d_{\delta' i} + d_j \quad (3.35)$$

In order for this label to be created, it has to be feasible, i.e.:

$$t_{\delta' j} \leq b_j \quad (3.36)$$

$$d_{\delta'j} \leq Q \quad (3.37)$$

In label correcting algorithms, labels are extended based on a procedure which scans all nodes iteratively. Each label ( $L_{\delta i}$ ) is extended to all other nodes and checked for feasibility. If not feasible, the new label is discarded. When a label  $L_{\delta i}$  has already been extended to all its feasible successor nodes, then it is considered as processed and can be deleted (or characterized as processed and kept for supporting dominance criteria, see below). The label extension procedure is repeated until there are no more unprocessed labels. Note that we have not yet considered the constraint to not re-visit the same vertex. This means that cycles are allowed and, therefore, the algorithm, so far, solves the non-elementary case. Infinite cycling is prohibited from the accumulation of resources (time and distance constraints).

When a partial path is extended to the ending node ( $n+1$ ) then a full feasible path has been generated. This path is a potential solution to the minimization problem. The minimum cost path among all feasible paths is the optimal solution.

Table 3.1 lists all the improvement procedures employed in our work, and identifies the relevant references in the literature. Note that the techniques mentioned in the following references constitute common improvement techniques that are employed by several authors, including Feillet *et al.* (2005). These improvement procedures are described below.

Table 3.1: Improvement Procedures for (E)SPPTWCC

Improvement Procedure	Source
Elementarity	Feillet <i>et al.</i> (2004), Chabrier (2006)
Buckets/Storing Processed Labels	Larsen (2001), Chabrier (2006)
Dominance Criteria	Dumas and Desrosiers (1986), Desrochers (1988), Feillet <i>et al.</i> (2004), Chabrier (2006)
Limited Discrepancy Search (LDS)	Feillet <i>et al.</i> (2005)
Preprocessing	Kontoravdis and Bard (1995), Desrochers <i>et al.</i> (1992)
Early Termination Criterion	Larsen (2001), Chabrier (2006)

### Elementarity

In order to extend labels strictly to nodes that have not yet been visited (elementary paths), Beasley and Christofides (1989) initially proposed the use of some additional elements in the labels. Feillet *et al.* (2004) were the first to implement this idea. Consider a vector  $R_{\delta i}$ , containing  $|N|$  binary elements (where  $|N|$  is the size of all nodes excluding the starting and ending ones), that represent partial route  $\delta$  ending at node  $i$ . All elements of  $R_{\delta i}$  initially are set to zero.

The new labels, proposed by Feillet *et al.* (2004), are  $L_{\delta i} = [\bar{c}_{\delta i}, t_{\delta i}, d_{\delta i}, R_{\delta i}]$ . When a label  $L_{\delta i}$  is extended to node  $j$ , the  $R_{\delta' j}$  vector of the new label  $L_{\delta' j}$  is equal to  $R_{\delta i}$  except from the  $j^{th}$  element of  $R_{\delta' j}$  which is set to 1. That is the case if  $j$  is visited for the first time in the partial path. In order to avoid re-visiting the same customers (elementarity), an additional feasibility check related to Eqs. (3.36) and (3.37) is included. That is, if node  $j$  has already been visited in the partial path  $\delta$  (i.e. the  $j^{th}$  element of  $R_{\delta i}$  is equal to 1), then label  $L_{\delta i}$  is not extended to node  $j$  (i.e. the new label  $L_{\delta' j}$  is not created).

### **Buckets (expand labels per each node)**

For each customer (node)  $i$ , there exists a set of non-processed labels,  $B(i)$ , which is called the *bucket* of node  $i$ . Every label in  $B(i)$  corresponds to a partial route that ends at node  $i$ .

Initially, only the set corresponding to the depot,  $B(0)$ , is non empty, containing one label,  $L_0$ . Then this label is extended to every node  $i$  creating partial paths  $(0 - i)$ . Thus, one label per node  $i$  is created and inserted in the corresponding  $B(i)$ . As mentioned above, labels that have been extended are characterized as processed and can be discarded. All newly created labels are tagged as non-processed. This is repeated for every  $B(i)$  in an iterative procedure until all labels have been processed. When there are no more unprocessed labels in all buckets the operation is terminated. Note that for node  $n + 1$ , all labels created are directly inserted in  $B(n + 1)$  if they satisfy the criterion of negative reduced cost, otherwise they are rejected.

### **Storing of Processed Labels**

In our algorithm we have followed the work of Chabrier (2006), in which labels that have been extended to all successors are kept in the set of processed labels  $\check{P}(i)$ , separately for each node  $i$ . Storing the labels in  $\check{P}(i)$  supports the solution process since these labels are (a) considered in the dominance checks, and (b) are useful in the LDS procedure discussed below.

### **Dominance Criteria**

Since the ESPPTWCC is NP-hard, many authors have developed dominance criteria that discard labels. Discarding labels improves the computational, as well as the memory, efficiency of the problem solved. The simplest dominance criterion is used by Dijkstra's algorithm for the solution of the single-source, single-destination SPP with positive arc costs. A label  $L_{\delta' i}$  dominates another label  $L_{\delta'' i}$  ending at the same node  $i$ , through different partial

paths  $\delta'$  and  $\delta''$ , respectively, when  $\bar{c}_{\delta'i} \leq \bar{c}_{\delta''i}$ . This process has proven to provide the optimal solution. In the present case and due to the additional resource consumption characteristics, this straightforward dominance criterion requires extension. Dominance criteria for the SPPTWCC were developed initially by Dumas and Desrosiers (1986), and Desrochers (1988) who proposed the following:

$$\bar{c}_{\delta'i} \leq \bar{c}_{\delta''i} \quad (3.38)$$

$$t_{\delta'i} \leq t_{\delta''i} \quad (3.39)$$

$$d_{\delta'i} \leq d_{\delta''i} \quad (3.40)$$

All three should be satisfied simultaneously and there should be at least one strict inequality. Although these criteria are appropriate for the SPPTWCC case, they are not sufficient for the elementary case, i.e. they do not guarantee optimality. Desrochers (1988), Feillet *et al.* (2004) and Chabrier (2006) have suggested an additional criterion for the case of the ESPPTWCC.

$$R_{\delta'i} \subseteq R_{\delta''i} \quad (3.41)$$

The basic idea is that a label that has visited or includes more nodes cannot dominate another label with fewer nodes, since the latter may lead to a better solution by visiting the customers that have already been included in the former. This additional dominance makes it harder for a label to dominate another, leads into maintaining numerous labels in each node, and, thus, increases complexity. Separately, Feillet *et al.* (2004) and Chabrier (2006) proposed modifications in order to reduce the search space (discard more labels) without sacrificing optimality.

In their procedure, the binary vector  $R_{\delta i}$  was modified in order to contain the *non-feasible* nodes (i.e. those that cannot be extended due to feasibility) in addition to the already visited nodes. Both these types of nodes are characterized as *unreachable* for the associated label  $L_{\delta i}$ . The inclusion of the non-feasible nodes in the vector  $R_{\delta i}$  provides a more robust implementation by discarding more unnecessary labels from the search space. That is, a label can discard another label, even if they have not visited the same customers, but both labels have the same successors. More robust dominance criteria related to Eq. (3.43) are provided by Chabrier (2006).

### Limited Discrepancy Search

Limited Discrepancy Search (LDS) was initially developed for Constraint Programming by Harvey and Ginsberg (1995). Feillet *et al.* (2005) successfully incorporated LDS into the solution procedure for the ESPPTWCC and, also, managed to solve several problem instances related to the Solomon benchmarks (1987) that were not solved previously.

Given a problem of  $|N|$  customers, a number of the  $m$  closest neighbors is defined, i.e. neighbors with the minimum modified arc cost ( $c'_{ij}$ ). The  $m$  closest customers to each node  $i$  are included in set  $G(i)$ . Note that the depot is always considered as a good neighbor. Also, for node 0 (i.e. the starting depot)  $m = |N|$ . Every label in  $B(i)$  and, thus, every partial route  $\delta$ , is characterized by a cumulative penalty. Extending a label to a node  $l$  that is not included in  $G(i)$  imposes a penalty ( $\gamma_{il}$ ) equal to 1, otherwise the penalty equals zero.

Initially, the allowable cumulative penalty ( $CP$ ) for a partial route  $\delta$  that corresponds to a label  $L_i$ , is set to zero and therefore labels are extended only to good neighbors. That is  $\sum_{(i,j) \in r} \gamma_{ij} = 0$  and therefore only arcs with  $\gamma_{ij} = 0$  are selected. After extending all labels, and if there are routes with negative costs, the ESPPTWCC terminates and passes these routes to the RMP. If there are no routes with negative cost,  $CP$  is increased by 1 and labels with  $\sum_{(i,j) \in r} \gamma_{ij} = 1$  are also allowed. An upper limit ( $CP_{limit}$ ) is defined, up to which  $CP$  can be increased. If the  $CP_{limit}$  has been reached by  $CP$ , and the subproblems have not generated any negative cost routes, then the operation terminates.

### Preprocessing

Large time windows increase the complexity of the problem, since more customers can be served by a single route and the possible combinations increase dramatically. For this reason, many researchers have suggested to use preprocessing procedures prior to starting the solution process in order to narrow the time windows of customers (without sacrificing feasibility), and, therefore, tighten the solution space.

Kontovardis and Bard (1995) proposed a simple criterion, in which each time window  $[a_i, b_i]$  of node  $i$  can be replaced by  $[\max(a_0 + t_{0i}, a_i), \min(b_{n+1} - t_{i,n+1}, b_i)]$ , where  $t_{ij}$  is the travel time between customers  $i$  and  $j$ , and  $[a_0, b_{n+1}]$  is the time window of the depot. That is, the starting time of a time window is set to the earliest time that a vehicle can reach the corresponding customer directly from the depot, and the ending time is set to latest time a vehicle can depart from the customer in order to arrive to the depot on time.

Desrochers *et al.* (1992) proposed an iterative procedure in order to further tighten the time windows. The procedure includes 4 rules. Given the time window  $[a_l, b_l]$  of customer  $l$ , and considering customers  $i$  and  $j$  as the predecessor and successor nodes of  $l$ , define the following:

- Minimal arrival time from the predecessors of node  $l$ :

$$a_l = \max \{a_l, \min\{b_l, \min_i\{a_i + t_{il}\}\}\}$$

- Minimal arrival time to successors:

$$a_l = \max \{a_l, \min\{b_l, \min_j\{a_j + t_{lj}\}\}\}$$

- Maximal departure time from predecessors:

$$b_l = \min\{b_l, \max\{a_l, \max_i\{b_i + t_{il}\}\}\}$$

- Maximal departure time to successors:

$$b_l = \min\{b_l, \max\{a_l, \max_j\{b_j - t_{lj}\}\}\}$$

These four rules are applied iteratively until there exist no more adjustments of the time windows. The second and third rules were also derived by Cyrus (1998). Note that these operations are feasible when the triangular inequality holds. This is the case here, since time is always increasing when customers are added to routes.

### Early Termination Criterion

Many researchers have proposed to terminate the subproblem solution procedure when a certain number of negative cost routes has been reached. Note that although this termination does not guarantee that the optimal solution to the subproblem has been reached, optimality of the global decomposition algorithm is still maintained. This is because: (a) the global algorithm iterates between RMP and the subproblems, and, eventually, the global optimum can be achieved, and (b) reaching the optimal solutions of the subproblems in each iteration is computational expensive.

Early termination is used by the majority of researchers dealing with the column generation procedure. Note that the efficiency of the early termination criterion with respect to different numbers of generated negative cost routes have been studied in Larsen (2001) for VRPTW instances.

### 3.4 COMBINING RMP WITH THE SUBPROBLEMS

Figure 3.2 illustrates the structure of the global column generation algorithm for the multi period routing problem. When solving an RMP, the associated shadow prices are generated, in addition to the actual solution (cost and relevant routes). These shadow prices are passed to the  $P$  subproblems, and are used to compute the modified costs  $c'_{ij}, \forall i, j \in N$ .

These costs are the elements of the cost matrix in the ESPPTWCC. Shadow prices  $\sigma_p$  are also incorporated in the cost matrix, through the starting depot's modified costs, thus  $c'_{0j} = c_{0j} - \sigma_p, \forall p \in P$ .

On the other hand, solving each subproblem generates a set of negative cost routes. Each of these routes is translated to a column, i.e. the equivalent representation of the route when variables  $a_{ir}^p$  variables are used. Note that the customer sequence of these routes has to be maintained separately. These routes are provided to the RMP and added to the existing routes – columns of the problem. The solution process terminates when no more routes with negative cost can be generated by any subproblem, which mirrors the classical termination procedure for the simplex method. In this case, the minimum cost solution from the last RMP is returned as optimal.

#### Notes on Complexity

The following should be noted regarding the complexity of the solution algorithm for the MPVRPTW:

- The Simplex method used for the RMP has an exponential worst-case complexity, although in practice performs efficiently for several cases (Papadimitriou and Steiglitz, 1998).
- As stated in Chapter 2, the ESPPTWC is NP-Hard (Dror, 1994; Kohl, 1995).
- As noted by Kallehauge et al. (2005), the “behavior of the dual variables plays a pivotal role in the overall performance”; Which in accordance with the number of columns generated at each iteration of the Column Generation (Larsen, 2001), affects the coordination of the RMP and the ESPPTWCCs, and also the number of iterations.
- The Branch & Price method (see Chapter 5), although in practice performs better than exhaustive search, it has an exponential worst case complexity that is driven by the problem size.

From the above references, it is concluded that the complexity of the algorithm for the MPVRPTW is of exponential worst case complexity.

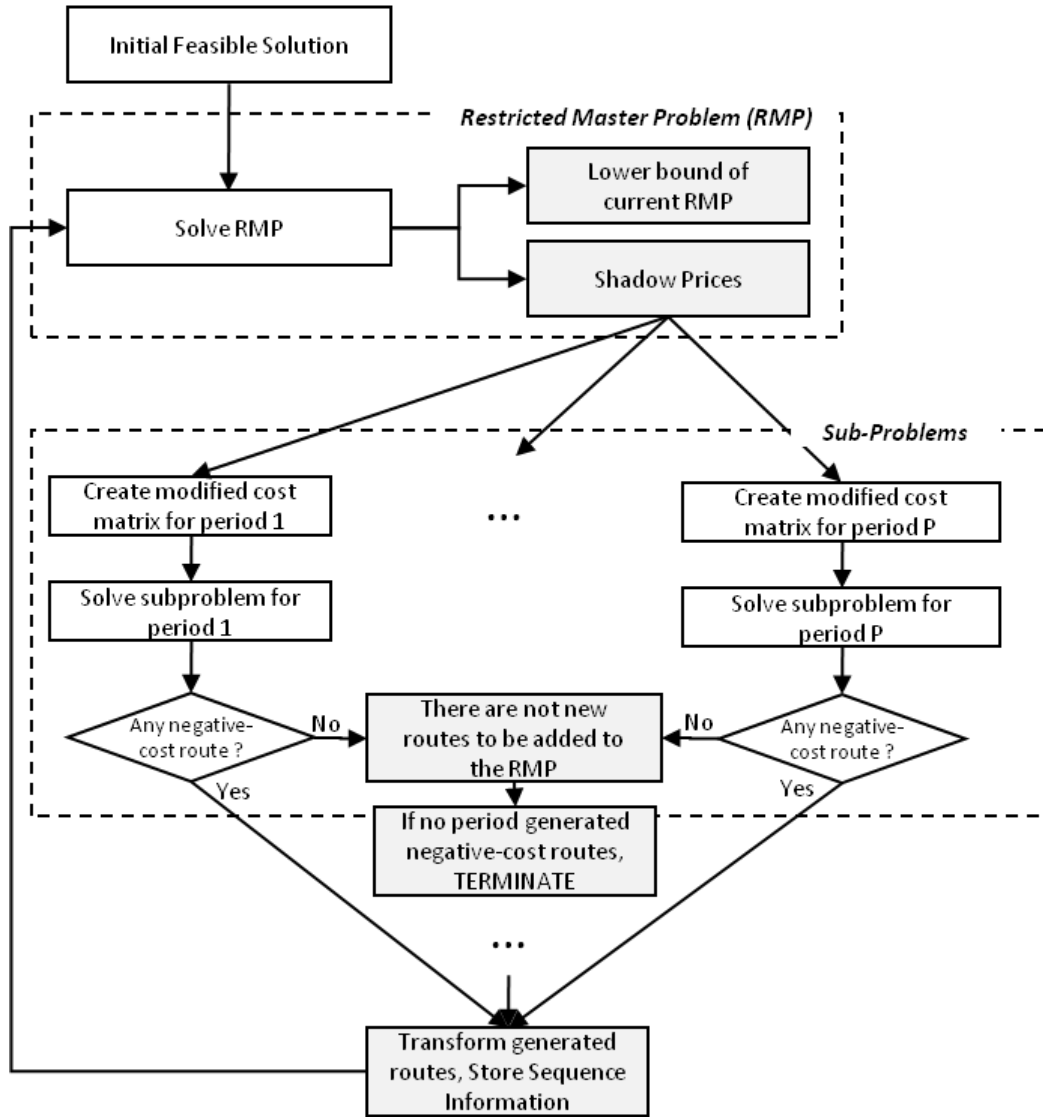


Figure 3.2: Column generation procedure for multi-period problems

### Optimality of Lower Bound

The Column Generation procedure solves the relaxed multi period routing problem optimally. Indeed, the solution procedure to the ESPPTWCC (See Section 3.3) provides optimal solutions to the subproblems and, thus, returns the columns (routes) with the minimum negative reduced cost. In case an optimal solution to the current RMP has been reached, the subproblems will not be able to return columns with negative reduced cost. Thus, there is no column (route) that exists in any  $\Omega_p$  that can further improve the lower bound.

Note that since the column generation procedure operates on the relaxed RMP, integer optimality is not guaranteed. In order to obtain the optimal integer solution, the column generation procedure is embedded in a Branch and Price framework of Chapter 5.



## Chapter 4:      ACCELERATING TECHNIQUES FOR THE MPVRPTW

In this Chapter, we propose alternative solution methods for the ESPPTWCC subproblems, exploiting the structure of the multi-period setting. As such, we focus on improving the solution framework presented in recent work to address different, but related, problems in the literature. The techniques presented in this Chapter improve the computation of the lower bound. In Chapter 5 we present enhancements regarding the evaluation of the optimal integer solution.

Considering the lower bound, two novel variations of the column generation method are developed targeting improved computational times and, thus, solutions of higher quality within a certain computational time period. In addition, the classical solution approach of Chapter 3 has been implemented in a parallel algorithm. All methods (i.e. the classical method, the two variations, as well as the parallel version) are compared in terms of computational times.

The proposed variations explore the solution space of the ESPPTWCC subproblems taking advantage of two major characteristics of these problems:

- (a) Flexibility of customers; customers are allowed to be routed in different periods. This may lead to routes that are common in different periods
- (b) Subproblem independence; each subproblem provides solutions that do not affect the other subproblems.

In order to illustrate the possible extend of common routes among periods, consider a problem with  $N$  customers and 2 periods. From these  $N$  customers, let the subset  $(N_1)$  contain all customers that can be routed in period 1, and the subset  $N_2$  contain all customers that can be routed in period 2. The common customers in these two subsets, which can be routed in both periods are noted as  $N_{1|2} = N_1 \cap N_2$ . Additionally let  $maxc$  be the upper bound of the number of customers that can be inserted in any route due to feasibility constraints. The number of common routes  $R_{1|2}$  in the two periods is given by the following equation:

$$R_{1|2} = \sum_{c=1}^{maxc} \binom{N_{1|2}}{c} = \sum_{c=1}^{maxc} \frac{N_{1|2}!}{(N_{1|2} - c)! c!} \quad (4.1)$$

Note that Eq. (4.1) does not consider the visiting sequence of customers, thus route [D-1-2-D] is the same as [D-2-1-D].

Figure 4.1 presents the ratio  $\frac{R_{1|2}}{R_1}$  of the sets  $N_1$  and  $N_{1|2}$  for 1 to 10 customers, taking  $maxc$  to be equal to the maximum number of customers per customer set. For example, for  $|N_1| = 3$  and  $|N_{1|2}| = 2$ , the ratio is:

$$\frac{R_{1|2}}{R_1} = \frac{\sum_{c=1}^{maxc} \frac{N_{1|2}!}{(N_{1|2} - c)! c!}}{\sum_{c=1}^{maxc} \frac{N_1!}{(N_1 - c)! c!}} = \frac{\sum_{c=1}^2 \frac{2!}{(2 - c)! c!}}{\sum_{c=1}^3 \frac{3!}{(3 - c)! c!}} = \frac{3}{7} = 43\%$$

Although, ratio  $R_{1|2}/R_1$  is below 25% in the most of the example cases, the proposed accelerating methods succeed in significant computational time reductions.

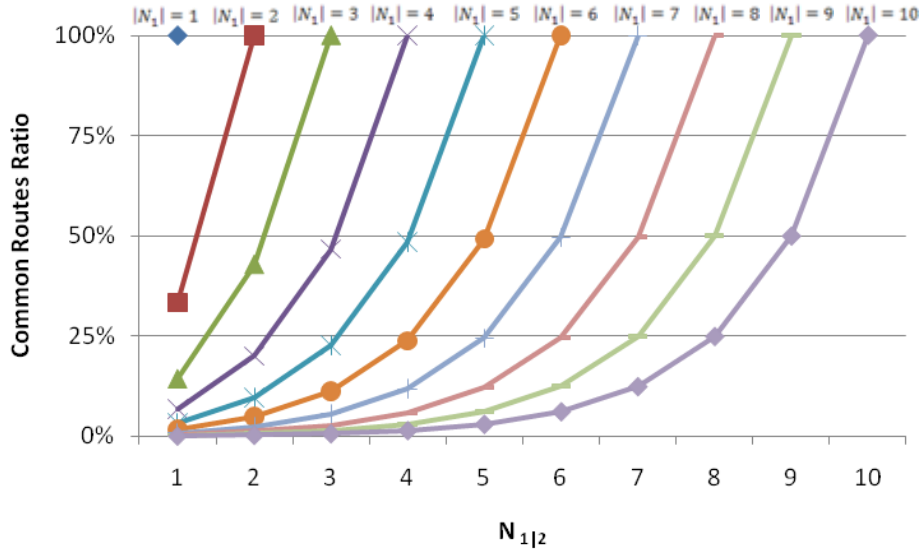


Figure 4.1: Common routes ratio for different sizes of set  $N_1$  and  $N_{1|2}$

Using the proposed methods we are trying to identify and exploit the common routes that are created among different periods, thus eliminating the computational effort needed to generate the same routes per period.

Below, each one of the proposed methods is presented highlighting its differences with the classical solution procedure (Chapter 3).

## 4.1 UNIFIED SUBPROBLEM METHOD

This acceleration method replaces the  $P$  subproblems of the classical solution method (Chapter 3) with a single (multi period) subproblem. The latter provides the necessary

columns (routes) to the RMP for all periods of the planning horizon. The idea behind this approach is that many of the partial paths, as well as final routes, that are generated during the solution of each subproblem (ESPPTWCC) are feasible in more than one periods. Thus, instead of solving separate subproblems to generate routes per period, we generate those by solving one common subproblem. The single subproblem is artificially constructed to take into consideration all customers within the planning horizon. Furthermore, it maintains additional information regarding the feasibility of the partial paths (or full routes) in the available periods of the planning horizon.

Note that the reduced cost of Eq. (3.21) includes the shadow price ( $\sigma_p$ ) which is relevant to period  $p$ . Since the subproblem is common for all periods, inclusion of the shadow prices ( $\sigma_p$ ) of each period  $p$  in the objective function is not possible. Thus, these shadow prices are eliminated from Eq. (3.21), forming the *relaxed* version of it:

$$\tilde{c}_r = \sum_{i \in N} a_{ir}^p c'_{ij} \quad (4.2)$$

Below we discuss all modifications to the standard approach that are proposed in order to implement the unified method.

### Modification of Labels

In order to consider period feasibility, each label  $L_{\delta i}$  of a partial path  $\delta$  ending at node  $i$  is modified by the addition of new elements. The new modified label  $L_{\delta i} = [\tilde{c}_{\delta i}, t_{\delta i}, d_{\delta i}, R_{\delta i}, \Phi_{\delta i}]$  includes an additional vector  $\Phi_{\delta i}$  of  $P$  binary elements ( $\varphi_{\delta i}^p$ ). Each of these elements is equal to 1 if label  $L_{\delta i}$  is feasible for period  $p$ , or 0 otherwise. The starting label  $L_0$  is feasible for all periods and, thus,  $[\varphi_0^1, \dots, \varphi_0^P] = [1, \dots, 1]$ . When extending label  $L_{\delta i}$  to a node  $j$ , vector  $\Phi_{\delta' j}$  for the new label  $L_{\delta' j}$  is given by the following equation:

$$\varphi_{\delta' j}^p = \begin{cases} \min(\varphi_{\delta i}^p, 1) & \text{if } p \in [\xi_j^s, \xi_j^e] \\ 0 & \text{else} \end{cases} \quad (4.3)$$

where  $[\xi_j^s, \xi_j^e]$  is the period window of customer  $i$ . Thus, each label is associated to the periods comprising the period window of each customer.

### Label Feasibility (for partial paths)

In the Unified method feasibility considers vector  $\Phi_{\delta i}$ . A label is feasible if Eqs. (3.36) to (3.37) hold and, additionally if the corresponding path ( $\delta$ ) can be routed in at least one period. That is, if at least one element  $\varphi_{\delta i}^p = 1$ . If  $\varphi_{\delta i}^p = 0, \forall p \in P$  then the associated label  $L_{\delta i}$  can

be eliminated, since it is infeasible for every period and cannot be extended to other feasible labels. In that case, label  $L_{\delta i}$  can be fully discarded.

### **Solution Feasibility (for final routes)**

Keeping a label  $L_{\delta, n+1}$  for a route that reached the ending depot is not as straightforward as in the classical solution procedure (Section 3.2), due to the exclusion of the shadow prices ( $\sigma_p$ ) from the *relaxed* reduced cost equation. As mentioned above, shadow prices  $\sigma_p$  have not been considered in the relaxed reduced cost ( $\tilde{c}_{\delta, n+1}$ ) of a label  $L_{\delta, n+1}$ , which has been accumulated up to node  $n + 1$ . Note also that based on these shadow prices, a feasible label  $L_{\delta, n+1}$  with negative *relaxed* reduced cost ( $\tilde{c}_{\delta, n+1}$ ), may not be of negative (actual) reduced cost ( $\bar{c}_{\delta, n+1}$ ) for every period  $p$ . In this case, for each label  $L_{\delta, n+1}$ ,  $P$  different reduced costs are calculated, one per period. Each of these reduced costs considers the shadow price  $\sigma_p$  of the relevant period  $p$ :

$$\bar{c}_{\delta, n+1}^p = \tilde{c}_{\delta, n+1} - \sigma_p \quad \forall p \in P \quad (4.4)$$

These reduced costs  $\bar{c}_{\delta, n+1}^p$  are stored separately. For every  $p \in P$  and for each label  $L_{\delta, n+1}$  for which  $\bar{c}_{\delta, n+1}^p \geq 0$ , the relevant  $\varphi_{\delta, n+1}^p$  is updated to 0, in order to exclude it from the feasible routes of period  $p$ .

Based on the above formulation, each label  $L_{\delta, n+1}$  with  $\tilde{c}_{\delta, n+1} \geq 0$  is discarded immediately, since it will remain positive for every  $p \in P$  (note that  $\sigma_p \leq 0, \forall p \in P$ ). A stricter bound is to eliminate labels  $L_{\delta, n+1}$  for which the following holds (this bound is used in our implementation):

$$\tilde{c}_{\delta, n+1} - \max_p(\sigma_p) \geq 0 \quad (4.5)$$

### **Dominance Criteria**

For the Unified method, in addition to the dominance criteria (presented in Section 3), the following should also hold, in order for a label  $L_{\delta' i}$  to dominate another label  $L_{\delta'' i}$  ending at the same node  $i$ :

$$\varphi_{\delta' i}^p = \varphi_{\delta'' i}^p = 1 \quad \forall p \in P \quad (4.6)$$

That is, labels can be checked for dominance only for the periods for which both labels been compared are feasible. Note that Eq. (4.6) should be checked for every period  $p \in P$ . That is,

for each label pair,  $P$  dominance checks are performed. If a label  $L_{\delta'i}$  dominates another label  $L_{\delta''i}$  for a period  $p$ , then, instead of eliminating  $L_{\delta''i}$ , the element  $\varphi_{\delta''i}^p$  relevant to period  $p$  is set to zero. Thus, label  $L_{\delta'i}$  is maintained for all other periods, while it will not be extended for period  $p$ .

The following property is significant in the Unified method.

The dominance criteria of Eq. (3.38) are valid when using the relaxed reduced costs without considering the relevant periods and the associated shadow prices.

*Proof:* In order for a label  $L_{\delta'i}$  to dominate another label  $L_{\delta''i}$  for period  $p$ , Eq. (3.38) should hold. That is,

$$\bar{c}_{\delta'i}^p \leq \bar{c}_{\delta''i}^p \Rightarrow \tilde{c}_i' - \sigma_p \leq \tilde{c}_i'' - \sigma_p \Rightarrow \tilde{c}_i' \leq \tilde{c}_i'' \quad (4.7)$$

Thus, since the shadow prices have been eliminated from the equation, it holds  $\forall p \in P$  and, therefore the dominance criteria using the relaxed reduced costs are applicable.

### Limited Discrepancy Search (LDS)

Working with a single subproblem, instead of  $P$  independent ones, affects also the implementation of LDS. In the classical implementation, for every customer  $i$  the  $m$  closest customers are selected and set as "good neighbors",  $GN(i)$  for every  $p \in P$ . This allows a directed search to be implemented over the node graph of the most promising arcs. Since the concept of different periods is not present in the Unified method, sets  $GN(i)$  are defined over a different customer set.

Consider a case in which customers  $i$  and  $j$  are close but cannot be routed in the same period. Since in the Unified strategy all customers are considered jointly, LDS would have included customer  $j$  into  $GN(i)$ , thus creating an infeasible connection. In order to avoid this,  $GN(i)$  is defined to include only the customers that can be routed in the periods in which customer  $i$  is feasible, i.e. periods  $[\xi_i^S, \xi_i^E]$ .

For example, consider customers  $i, j$  and  $k$  with period windows  $[1,2]$ ,  $[2,3]$  and  $[3,4]$ , respectively (see Fig. 4.2). If only one good neighbor is allowed per customer, then in the case of  $G(j)$ , this allowable neighbor will be selected among customers  $i$  and  $k$  in the Unified strategy. In contrast, in the classical method the good neighbors of customer  $j$  will be defined per period (i.e. per subproblem) and, thus, for period 2, the good neighbor would be customer  $i$  and for period 3 it would be customer  $k$ .

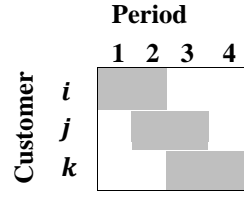


Figure 4.2: Feasible periods per Customer

### Termination Criteria

We terminate the solution procedure of the Unified method when at least 500 feasible columns (routes) with negative reduced cost have been determined for any of the periods. Note that at any point during the solution procedure, vector  $\Phi_{\delta,n+1}$  holds the information regarding period feasibility of each label  $L_{\delta,n+1}$ . The calculation of the number of the final negative cost solutions per period is performed using the information in these vectors.

## 4.2 CLONING METHOD

The Cloning method exploits further the customer flexibility intrinsic to this problem, i.e. the flexibility of customers to be routed in alternative periods of the planning horizon. Similar techniques to the Cloning method have been mentioned by Pirkwieser and Raidl (2009) and Mourgaya and Vanderbeck (2007) for the PVRP.

The key idea is to select and solve only a subset of the subproblems, called hereafter the *parent* subproblems, and transfer a selected feasible part of their solution (e.g. columns/routes) to the remaining subproblems (hereafter called the *linked* subproblems). More specifically, consider  $P$  subproblems, one for each period. We select  $P' (\subseteq P)$  subproblems to solve. To each *parent* subproblem  $p' \in P'$ , there is a set of *linked* subproblems  $LP_{p'}$ , such as  $P' \cup (\bigcup_{i \in P \setminus P'} LP_i) = P$ . Routes generated by each  $p' \in P'$  are considered for inclusion in their linked subproblems  $LP_{p'}$ . Note that these routes are feasible in terms of time windows, capacity, elementarity, etc. but they may contain customers not allowed to be included in the *linked* subproblems. If the solution of a parent subproblem  $p' \in P'$  generates at least one feasible column (route) for a subproblem  $p''$  from the subset  $LP_{p'}$ , then the latter is considered as solved, i.e. it is not solved separately. These columns are also included in the RMP for period  $p''$ . Thus, the explicit solution of every subproblem is avoided. It is noted, however, that in the last iteration of the method, all subproblems will be solved separately in order to secure optimality.

Finally, note that since only a subset of the subproblems are solved explicitly, different or fewer columns are returned for the *linked* subproblems. Therefore, this method traverses through different extreme solutions of the RMP's convex hull as compared to the classical approach.

Significant issues of the Cloning method are discussed below.

### Cloning routes to other subproblems

Consider two periods,  $p_1$  and  $p_2$  with shadow prices  $\sigma_{p_1}$  and  $\sigma_{p_2}$ , respectively, where the following holds:  $\sigma_{p_1} \leq \sigma_{p_2} \leq 0$ . Note that  $\sigma_p \leq 0, \forall p \in P$ . Solving the subproblem for period  $p_1$  generates a set of feasible columns (routes) for this period. In order to determine if these routes can be included in period  $p_2$ , in addition to feasibility their reduced cost should be negative. This is checked by recalculating their reduced costs as follows:

$$\bar{c}_r^{p_2} = \bar{c}_r^{p_1} + \sigma_{p_1} - \sigma_{p_2} \Rightarrow \bar{c}_r^{p_2} = \bar{c}_r^{p_2} - \sigma_{p_2} \quad (4.8)$$

Only the routes with negative reduced cost ( $\bar{c}_r^{p_2} < 0$ ) and which are feasible in period  $p_2$  are maintained and included in the RMP for period  $p_2$ .

### Optimality

Not solving explicitly all subproblems may lead to suboptimal solutions. There are two main considerations that should be investigated concerning optimality:

#### Consideration 1: Solution of all subproblems in the final iteration

The parent subproblems may not generate feasible columns (routes) for all linked subproblems. Thus, it is necessary to solve each subproblem explicitly in the final iteration.

*Justification:* Continuing the previous example, every generated solution will have reduced cost equal to  $\bar{c}_r^{p_1} = \bar{c}_r^{p_1} - \sigma_{p_1}$  where  $\bar{c}_r$  is the relaxed reduced cost (as described in Section 4.1). Considering only the negative cost solutions and given that  $\bar{c}_r^{p_1} < 0 \Rightarrow \bar{c}_r^{p_1} < \sigma_{p_1}$ ; that is, the generated routes have relaxed reduced cost ( $\bar{c}_r^{p_1}$ ) lower than  $\sigma_{p_1}$ . However, if the shadow price  $\sigma_{p_2}$  of a linked subproblem is larger ( $\sigma_{p_1} < \sigma_{p_2}$ ), then those feasible routes of subproblem  $p_2$  with negative-cost in the interval  $(\sigma_{p_1}, \sigma_{p_2})$  will not be generated by the solution of subproblem  $p_1$ . This is illustrated in Fig. 4.3, in which the costs of all columns (routes) generated by the subproblem of period  $p_1$  are located in interval A. Routes with costs inside interval B will be ignored by this subproblem as non negative. Interval C contains routes with positive reduced cost for both periods (subproblems).



Figure 4.3: Illustrative example of the Cloning strategy suboptimality

Thus, in case a parent subproblem does not generate any feasible route for a linked subproblem, this linked subproblem should be solved independently in order to guarantee optimality.

It is noted that maintaining all routes with reduced cost less than  $\sigma_{p_2}$  would have included all feasible solutions of period 2, but would also have increased considerably the computational time for solving the subproblem of period 1.

#### Consideration 2: Dominance Criteria Validation

The dominance criteria are valid for the routes of both the parent and the linked subproblems. Thus, (a) additional criteria are not needed, and (b) all feasible routes related to the linked subproblems (with reduced cost less than  $\sigma_{p_1}$ , as discussed above) will be generated.

*Justification:* Consider two labels,  $L_{\delta'i}$  and  $L_{\delta''i}$ , ending at the same node  $i$ , and two periods: Period  $p'$  related to the parent subproblem, and period  $p''$  related to the linked one. Consider the following two cases:

- (a) *Label  $L_{\delta'i}$  is feasible only in period  $p'$  and label  $L_{\delta''i}$  is feasible in periods  $p'$  and  $p''$ .*

In order for the former label to dominate the latter, Eq. (3.41) should hold, that is  $R_{\delta'i} \subseteq R_{\delta''i}$ . Since label  $L_{\delta'i}$  is feasible only in period  $p'$ , it includes at least one customer that cannot be serviced in period  $p''$ , thus, Eq. (3.41) is not valid and label  $L_{\delta''i}$  will not be discarded.

- (b) *Both labels are feasible in periods  $p'$  and  $p''$ .* In this case, both labels can lead to feasible solutions to the linked subproblem. If label  $L_{\delta''i}$  is dominated, then it can be discarded since its successor labels will always be dominated by the successors of label  $L_{\delta'i}$ . That is also valid if the linked subproblem was solved explicitly.

Thus, all feasible solutions to a linked subproblem, with reduced cost less than  $\sigma_{p_1}$ , may be generated by its parent subproblem.

### 4.3 IMPLEMENTATION OF PARALLEL SOLUTION

In this case we solve the  $P$  subproblems of the classical solution procedure (Section 3.2) in parallel. Consider Fig. 4.4 which presents the sequential and the parallel procedures. In both cases, if there are negative-cost columns generated by any of the subproblems, then the RMP and the subproblems are solved again.

For the parallel implementation we used an 8-core Windows XP machine (with 8 matlab workers, i.e. parallel processors). Since in our case the number of parallel processors is greater than the maximum number of periods ( $P = 5$ ), it is possible to distribute all subproblems to the independent processors. The minimum computational time to solve the subproblems lies between  $\max_p(t_p)$  and  $\sum_{p=1}^P t_p$  (of the sequential case), where  $t_p$  is the computational time to solve the subproblem of period  $p$ . Note that in practice the minimum computational times cannot be reached due to computational overheads (such as distributing, collecting and merging variables and data to/from the processors).

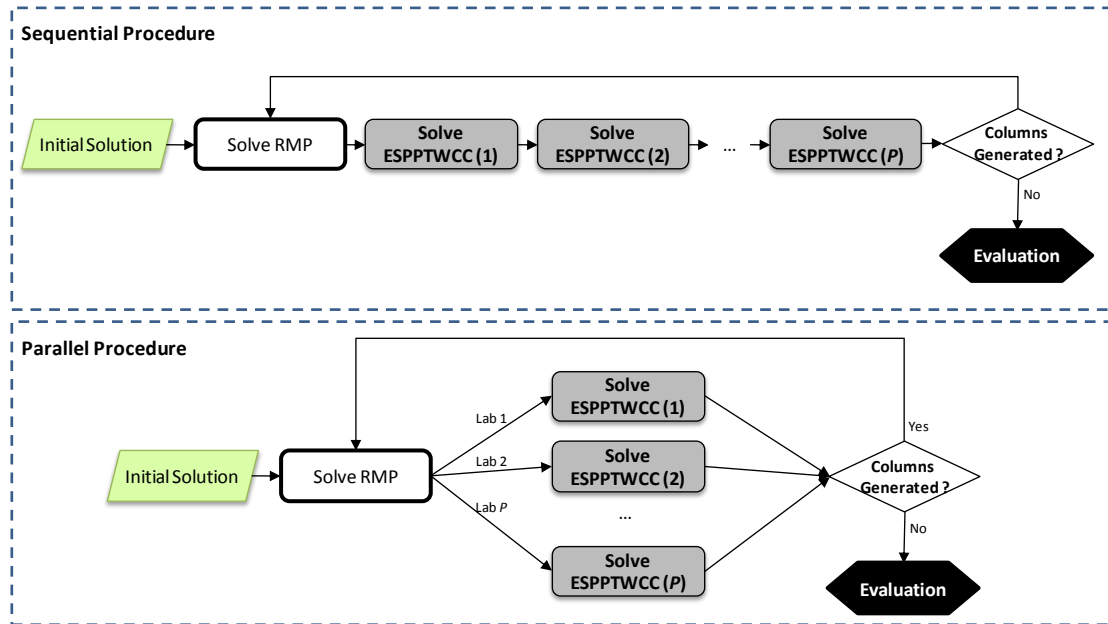


Figure 4.4: Sequential procedure of the classical column generation and the parallel approaches for the Multi-period problem.

### 4.4 COMPUTATIONAL TECHNIQUES FOR SOLVING THE SUBPROBLEMS

#### Vectorization of looping procedures (3D vertices)

In order to (a) fully exploit the strength of the Matlab<sup>®</sup> software, i.e. to use vectorized operations, and (b) avoid its weaknesses, i.e. loop procedures (such as for, while, etc), we

used matrix operations to expand each bucket  $B(i)$ , i.e. the set of all labels ending at customer  $i$ . Note that matrices in Matlab<sup>®</sup> are stored as vectors in continuous memory space and can efficiently be managed as vectors.

A common (but expensive) approach to expand each label in a bucket would be to proceed with two embedded for-loops, as shown in Fig. 4.5:

---

```

for each label in B(i)
  for each node j
    if feasibility constraints are satisfied (i.e.,  $j \neq i$ ,  $j$  has not been visited, etc)
      expand label to node j
    end
  end
end
end

```

---

Figure 4.5: Pseudocode for label extension using looping procedures

In order to vectorize the label extension code, each set  $B(i)$  is represented by a two-dimensional matrix that includes the information related to all non-processed labels. Every row of  $B(i)$  corresponds to a label  $L_{\delta i}$  and contains all information pertinent to this label (Fig. 4.6). Thus, the matrix of set  $B(i)$  is of size  $Q \times M$ , where  $Q$  is the number of the labels contained in  $B(i)$  and  $M$  is the number of items that are stored in each label.

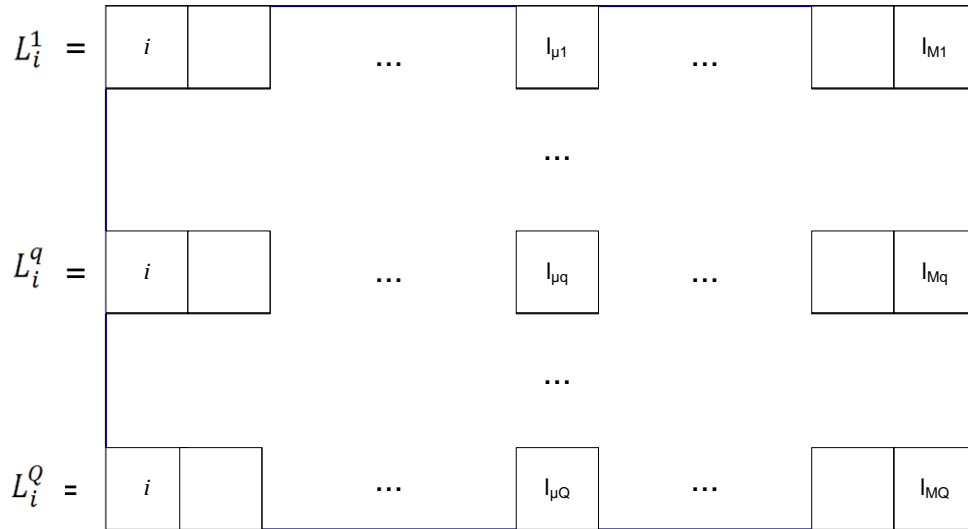


Figure 4.6: Set  $B(i)$  representation in a 2-D matrix.

Each label is extended to all other nodes. The newly created labels are kept in a three-dimensional matrix,  $S(i)$ , of size  $M \times Q \times N$ , where  $N$  is the size of the customer set.

Figure 4.7 presents an illustrative example of  $S(i)$ . Note that each label from set  $B(i)$ , when extended, creates labels that are stored in the  $M \times N$  part (grey area) of  $S(i)$ . Infeasible labels are not loaded to the related buckets  $B(i)$ . Feasibility takes under consideration elementarity, time windows, capacity and the unreachable nodes.

### Successors

A drawback of the vectorized loop procedures is that all matrices should be compatible in size in order to perform classical algebraic operations. Therefore, we cannot reduce the size of  $S(i)$  based on the feasible successor list per each label, in order to achieve faster computational times and less memory utilization. A successors list of a label  $L_i$  contains all other customers that this label can be extended to, that is, all customers for which vector  $R_i$  is equal to zero (i.e. is not unreachable).

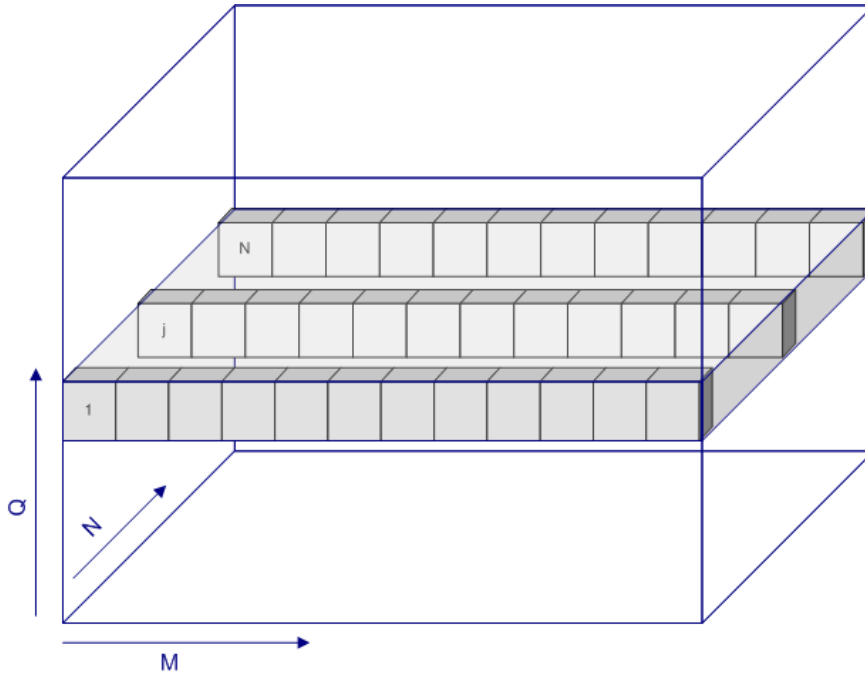


Figure 4.7: Set  $S(i)$  representation in a 2-D matrix.

For example, let  $B(i)$  contain only two labels,  $L_{\delta'i}$  and  $L_{\delta''i}$ . Suppose also that  $L_{\delta'i}$  cannot be extended to customer  $j$  and  $L_{\delta''i}$  cannot be extended to customers  $j'$  and  $j''$  (for feasibility reasons). Since customer  $j$  is unreachable (note that the information is kept in the *unreachable* nodes vector  $R_{\delta'i}$ ) from label  $L_{\delta'i}$ , there is no need to extend label  $L_{\delta'i}$  to customer  $j$ . Thus, the dimension  $N$  of  $S(i)$  could be reduced by 1. Following the same argument for label  $L_{\delta''i}$ ,

$N$  could be reduced by 2. However, this would result in a conflict regarding the size of  $S(i)$ . To address this issue, a different approach is used in order to create a common successor list for all labels in each  $B(i)$ .

Based on the information maintained in the vector of *unreachable* nodes, if a node  $j$  is unreachable for all labels of  $B(i)$ , then node  $j$  can be discarded from the label extension process. As a result, dimension  $N$  is reduced by one. Considering all common unreachable nodes of  $B(i)$ , size  $N$  is reduced accordingly. This operation leads to less memory utilization and faster matrix operations.

### **Dominance Criteria**

Newly created labels are checked using dominance criteria in a two-stage procedure. Consider a newly created label  $L_{\delta i}$  in  $B(i)$ . This is compared against (a) the remaining non-processed labels in  $B(i)$ , and (b) the processed labels in  $P(i)$ . A circular process is used as follows: Initially a label  $L_{\delta i}$  is checked if it dominates, or is dominated by, other labels within bucket  $B(i)$ . If label  $L_{\delta i}$  is not dominated by any label within bucket  $B(i)$ , it is checked if it dominates, or is dominated by, labels within the bucket  $P(i)$ .

In this process a new label can eliminate labels from sets  $B(i)$  and  $P(i)$ , and can also be eliminated by the labels in these sets. When a non-processed label  $L_{\delta i}$  is eliminated by a label within  $P(i)$ , then label  $L_{\delta i}$  is not extended further, contributing to computational time reduction. If every label was only checked with the non-processed labels (i.e. set  $B(i)$ ) at every iteration, then numerous labels would have been extended that are not needed. Maintaining sets  $P(i)$  eliminates these labels. Eliminating labels from  $B(i)$ , using either the labels within  $B(i)$  or within  $P(i)$ , minimizes the number of labels to be extended and, therefore, reduces computational time of the dominance procedure.

Matrix operations were also used for the implementation of the dominance criteria. In these operations, it is possible to reduce dimension  $M$  of  $B(i)$  or  $P(i)$ , as in the label extension process (see *vectorization of loop procedures*), by eliminating the common unreachable nodes of all labels within  $B(i)$  (or  $P(i)$ ). Unfortunately, the frequent iterative call of the dominance procedure may lead to the opposite results. The calculation of the common *unreachable* nodes, in each iteration, consumes more computational time than the time savings coming from the computational operations with reduced matrix sizes. In our implementation, these buckets are reduced based only on the unreachable nodes that stem from the preprocessing phase (See Chapter 3.3).

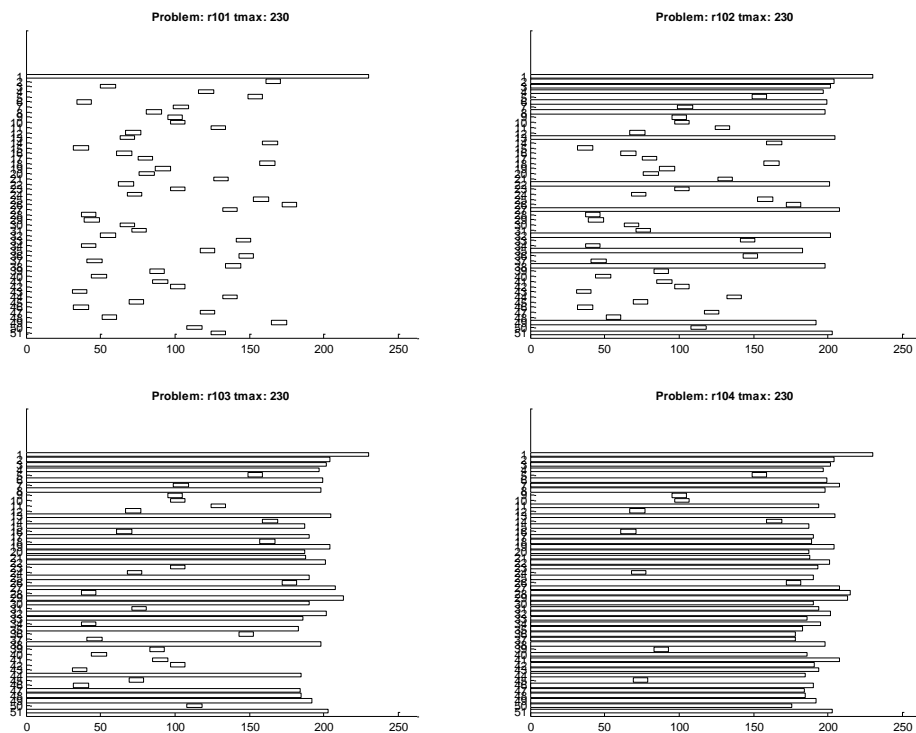
## 4.5 TEST INSTANCES AND BENCHMARK RESULTS

The aforementioned methods were tested and compared to the classical column generation method. For this purpose, a number of original test instances for the MPVRP were generated and are described below. In the following, the results of each method are labeled as follows: Classical column generation procedure (FULL), Unified method (UNI), Cloning method (CLONE), and Parallel method (PARA).

### 4.5.1 TEST INSTANCES

#### The Solomon Benchmarks

The original test instances for the MPVRP were created based on the Solomon Benchmarks. The latter comprise 6 problem sets (R1, C1, RC1, R2, C2, RC2), with each letter representing a different geographical distribution of customers (R: random, C: Clustered, RC: mixed). Each problem set comprises multiple problem instances; for example, the R1, C1 and RC1 problem sets comprise 12, 9 and 8 test instances, respectively. Furthermore, in each problem set, significant characteristics of the test instances, i.e. the number of clients (100), the customer coordinates, the demand, and the service times are identical. The difference between the instances in a set is the “tightness” of the customer time windows. For the R1 set, Figures 4.8 and 4.9 illustrate the different time windows per test instance.



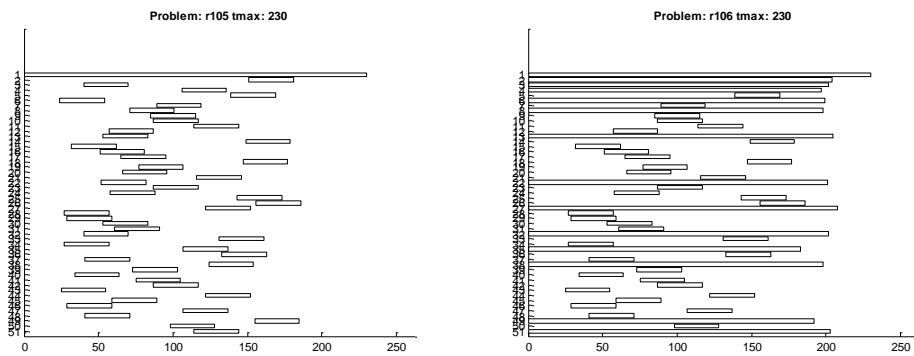


Figure 4.8: Time windows of R101 – R106 instances

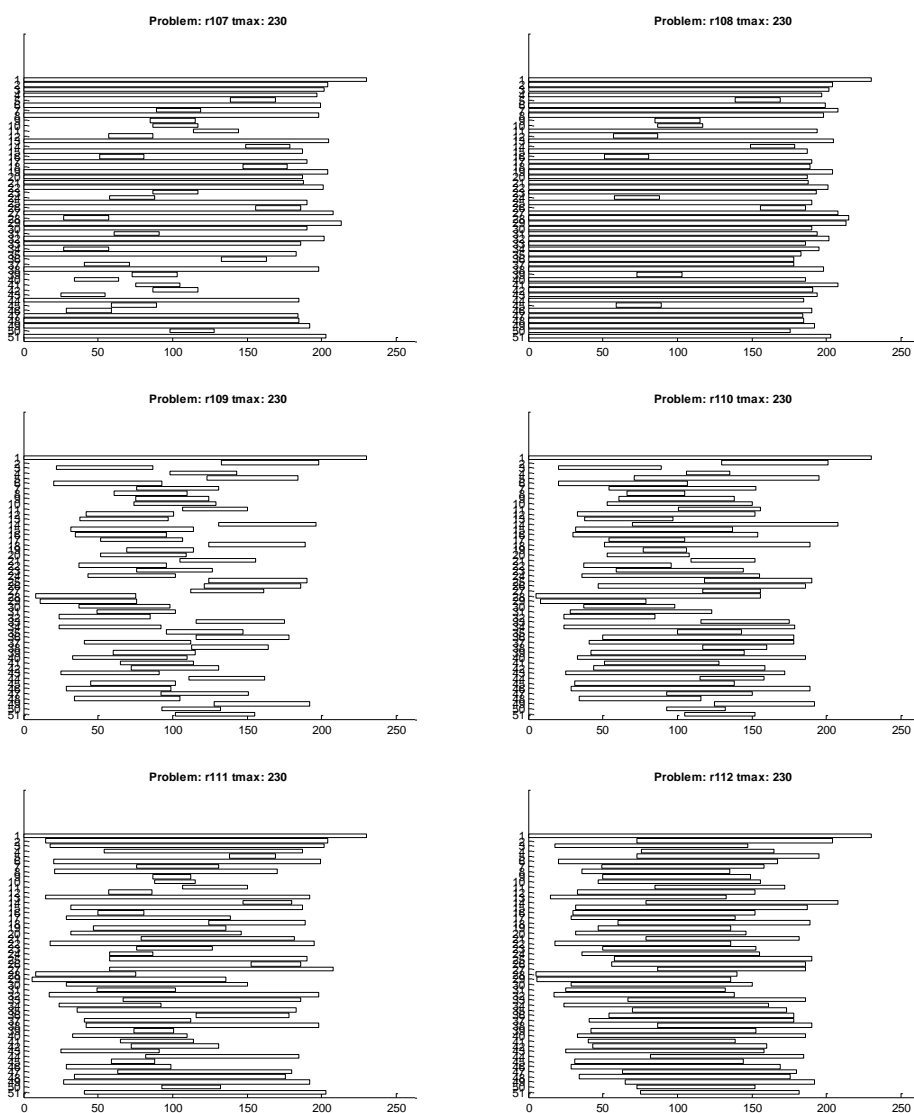


Figure 4.9: Time windows of the R107 – R112 instances

The example above shows that the various instances correspond to different time window patterns, ranging from narrow to wide time windows for most customers. In all cases, however, there are customers with time windows that extend to almost the entire time period. Instances R101 to R104 contain customers with narrow time windows, mixed with customers with wide time windows in different proportions. Instances R105 to R108 present the same pattern but with larger time windows. Instances R109 to R111 include customers with almost equal time windows. R112 presents a special case of large time windows. In this case, the majority of the time windows are placed in the middle of the available time period.

The average length of the time windows of all customers included in each problem instance of sets R1, C1 and RC1 are presented in Table 4.1, both in absolute units and as a percentage (%) of the maximum allowable time period.

Table 4.1: Average time windows per test instance (maximum allowable time: 230, 1236 and 240 for the R1, C1 and RC1 instances, respectively)

Probl.	Average TW/Cust.	% of Tmax	Probl.	Average TW/Cust.	% of Tmax	Probl.	Average TW/Cust.	% of Tmax
<b>R101</b>	10,00	4%	<b>C101</b>	60,76	5%	<b>RC101</b>	30,00	12%
<b>R102</b>	57,39	25%	<b>C102</b>	325,69	26%	<b>RC102</b>	71.08	30%
<b>R103</b>	102,99	45%	<b>C103</b>	588,49	48%	<b>RC103</b>	109.80	45%
<b>R104</b>	148,31	64%	<b>C104</b>	852,94	69%	<b>RC104</b>	156.54	65%
<b>R105</b>	30,00	13%	<b>C105</b>	121,61	10%	<b>RC105</b>	56.38	23%
<b>R106</b>	72,39	31%	<b>C106</b>	156,15	13%	<b>RC106</b>	60.00	25%
<b>R107</b>	112,99	49%	<b>C107</b>	180,00	15%	<b>RC107</b>	88.10	37%
<b>R108</b>	153,31	67%	<b>C108</b>	243,28	20%	<b>RC108</b>	111.62	47%
<b>R109</b>	58,89	26%	<b>C109</b>	360,00	29%			
<b>R110</b>	86,50	38%						
<b>R111</b>	93,10	40%						
<b>R112</b>	117,64	51%						

### Generation of MPVRP benchmarks

Multi-period benchmark test instances with 50 customers were created based on the R1, C1 and RC1 Solomon benchmarks. In order to transform the latter to multi-period problems, we have introduced a period-window for each customer as follows:

- The planning horizon is set to five (5) consecutive periods
- For each Solomon instance, the first 50 customers were selected and separated into 5 groups (10 customers per group in a sequential manner). Each of these groups was assigned a different period-window.

- Nine period-window patterns were developed in order to simulate different multi-period situations. Thus, for each Solomon test instance, nine different instances for the MPVRP were generated.

Table 4.2 presents the period windows per pattern for each group of ten customers.

Table 4.2: Period-window patterns

Group	Customers	Pattern								
		1	2	3	4	5	6	7	8	9
1	1 to 10	[1,1]	[1,1]	[1,1]	[1,1]	[1,1]	[1,2]	[1,3]	[1,4]	[1,5]
2	11 to 20	[2,2]	[1,2]	[1,2]	[1,2]	[1,2]	[1,2]	[1,3]	[1,4]	[1,5]
3	21 to 30	[3,3]	[2,3]	[1,3]	[1,3]	[1,3]	[1,3]	[1,3]	[1,4]	[1,5]
4	31 to 40	[4,4]	[3,4]	[2,4]	[1,4]	[1,4]	[1,4]	[1,4]	[1,4]	[1,5]
5	41 to 50	[5,5]	[4,5]	[3,5]	[2,5]	[1,5]	[1,5]	[1,5]	[1,5]	[1,5]

Figure 4.10 illustrates the differences among the nine patterns. In this Figure, the grey areas represent the period windows per pattern and per customer group.

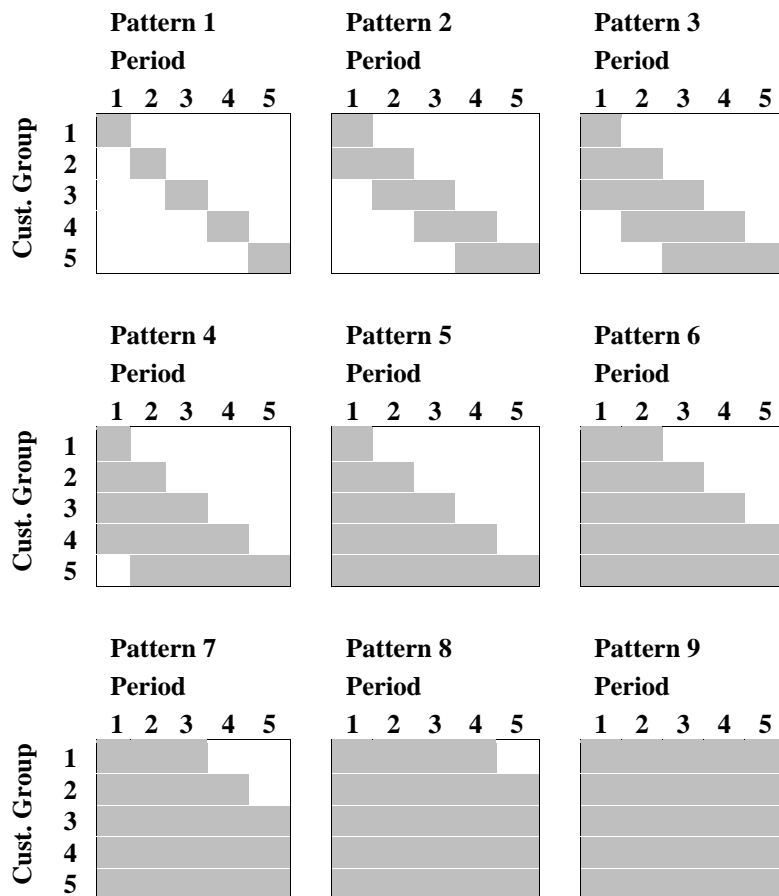


Figure 4.10: Period window patterns

These patterns represent different *degrees of customers' flexibility*. Patterns 1 and 9 can be considered as extreme cases for the multi-period problem: Pattern 1 allows no flexibility (the

period window length equals one period), while Pattern 5 allows full flexibility (period window length covers all periods). Thus,

- Pattern 1 can be solved by five independent VRPTWs, and
- Pattern 9 can be solved either using one single subproblem (in this case the  $P$  subproblems are identical), or by solving a VRPTW with available vehicles equal to the sum of the vehicles per period [i.e. solving a VRPTW with  $K = \sum_{p \in P} (K_p)$ ]. In this way, the final solution will result in a number of routes which can be distributed arbitrarily to periods, since all customers are feasible to all periods. Of course, none of the other patterns can be directly solved using the VRPTW formulation.

Taking into account the nine MVPVRP instances for each Solomon instance, 261 multi-period test instances are defined as follows:

- From the R1 set: 12 test instances x 9 patterns = 108 MPVRP instances
- From the C1 set: 9 test instances x 9 patterns = 81 MPVRP instances
- From the RC1 set: 8 test instances x 9 patterns = 72 MPVRP instances

All solution methods were applied to each of the 261 instances to obtain the optimal solution of the relaxed problem (lower bound). For each problem, the optimal solution was found by each method spending, of course, different computational effort (time). Thus, the analysis of the results concerns the efficiency (in computational time) of the alternative solution methods with respect to the following characteristics: (a) customer geographical distribution, (b) time window duration, and (c) period window pattern.

#### 4.5.2 INITIAL SOLUTION

Although an initial feasible solution is not necessary in the set-covering formulation, we provide one here, since it (a) helps the column generation procedure to converge faster, and (b) provides initial values of the shadow prices that are closer to their final optimal values. Note that in the case of unlimited fleet, a trivial initial solution can be used (e.g. one route per customer). In our case of limited number of vehicles, however, a solution needs to be constructed to respect the maximum number of vehicles. This initial feasible solution was generated based on the classical insertion algorithm suitably enhanced to accommodate the periodic characteristics of our problem, i.e.

1. All customers are sorted based on their period flexibility; i.e. customers with period windows that expire sooner are placed on top of the list. Customers with the same expiration period are sorted in descending order based on their distance from the depot
2. For every customer  $i$  a single visit route ( $r_i = D - i - D$ ) is defined along with its cost  $sc_i$ .
3. Starting with the most urgent and furthest from depot customer  $i$  (that is the customer on top of the list), the first actual route ( $r$ ) is defined for the first feasible period in the planning horizon (note that the customers are sorted based on their period window). The next customer say ( $l$ ) in the list is selected and entered into route  $r$  between the customers that define arc  $(i, j)$  with the lowest cost increase, that is:

$$\max_{(i,j) \in r} \{c_{ij} - (c_{il} + c_{lj}) + sc_l\} = \max_{(i,j) \in r} \{c_{ij} - (c_{il} + c_{lj}) + (c_{ol} + c_{lo})\} \quad (4.9)$$

4. When a route can no longer accommodate a customer  $l^*$  from the list, due to the time constraint, a new single visit route ( $r'$ ) is created using step (3) with the remaining unassigned customers
5. This process continues until the maximum number of available vehicles for the selected period has been reached, or all customers have been assigned to routes. If there are unassigned customers that cannot be served within the next periods, the procedure terminates without a feasible initial solution. Otherwise,
6. The next period is selected. Steps (3) to (5) are repeated for the selected period
7. The process terminates when either all customers have been assigned to periods and routes, or when there are unassigned customers that cannot be served (infeasible solution). In the latter case the operation terminates with no solution.

### 4.5.3 TEST RESULTS

The instances were solved using an 8-core Windows-XP machine. As already mentioned, the Parallel method used the default Matlab<sup>®</sup> parallel procedure (with 8 matlab workers, i.e. parallel processors).

With respect to the early termination criterion of the subproblems (Section 3.3), the maximum number of negative cost columns to be generated by each subproblem was set equal to 500. Analytical results are presented in Appendix A. Note that in the Unified method, termination occurs when at least 500 feasible columns have been found for any period.

In the Cloning method, the subproblem of the first period is solved initially. If it generates feasible columns to the subproblems of any following period, then these subproblems are not solved explicitly. In case there are no feasible columns for a period, then the subproblem of this period is solved and the generated columns are considered for possible linked subproblems. For every subproblem of period  $p$ , the subproblems of periods  $[p + 1, P]$  are considered as the linked subproblems.

In LDS: Parameter  $m$ , which defines the good neighbors was set equal to 10. Furthermore, the cumulative penalty ( $CP_{limit}$ ) was set equal to  $[10\% \times C_{max}]$ , where  $C_{max}$  is an upper bound on the maximum number of customers that can serviced by a route.  $C_{max}$  is equal to  $\min\{C_{max}^d, C_{max}^t\}$ , where (Feillet *et al.*, 2005):

- $C_{max}^d$  is the maximum number of customers that can be served by a vehicle without violating the vehicle capacity constraint. Defining as  $N'$  every subset of  $N$ :

$$C_{max}^d = |N'| : \max_{N'} \left\{ \sum_{i \in N'} d_i \right\} \leq Q$$

- $C_{max}^t$  is given by a similar formula with respect to the service time of each customer.

### Factor Analysis I: Time Windows, Period Window Patterns, Methods

Tests were conducted in order to evaluate the efficiency of each alternative solution method with respect to (a) the size of the time windows and (b) the different period window patterns. The test instances were separated into three categories based on the average time window length (see Table 4.1): Small (10% to 30% of the available time), medium (30% to 50%) and large (50% to 70%).

The results for each TW category (small, medium or large) are presented in Fig. 4.11. This figure presents, for each method, the cumulative computational time for all problems in a TW category. Note that, this cumulative time value refers to all problem sets (R1, C1, RC1) and all patterns.

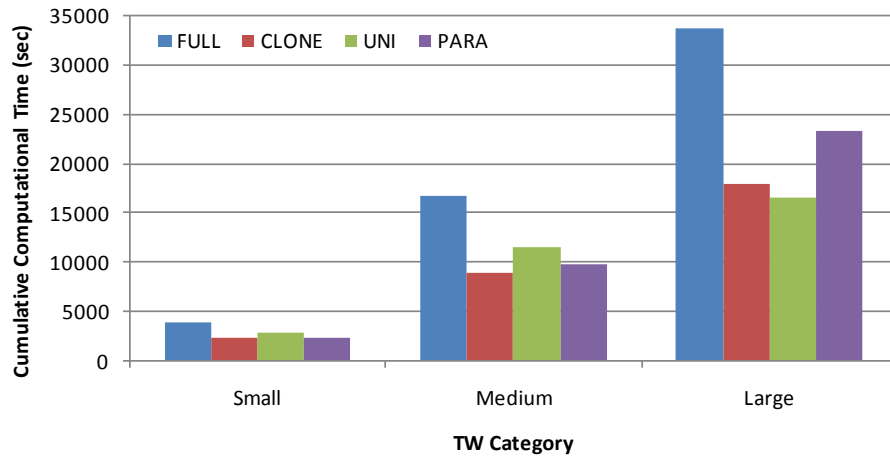


Figure 4.11: Cumulative computational time per method and TW category

As expected, instances with larger time windows result in increased computational times. In addition to this obvious observation, and as shown in Fig. 4.12, for all three TW categories the methods present similar behavior. That is:

- The classical solution method has the highest cumulative computational time among all solution methods
- As far as the remaining three methods, they succeed in reducing computational time by 36%, 40% and 43% on average, compared to the classical method, for the small, medium and large TW categories, respectively
- The Cloning method appears to be the most efficient method for small and medium time windows and is slightly outperformed by the Unified method for the large time window instances. The cloning method exhibits 46% time reduction on average in all three TW categories.
- The Unified method is the least efficient among the three alternatives for small and medium time windows. However, it appears to be the most efficient for large time windows, resulting in a time reduction of 51% compared to the classical approach.

Note that patterns with wider period windows are more computationally expensive, and this effect may not be apparent in the results of Fig. 4.11. To further investigate this, Table 4.3 and Fig. 4.12 present results on total computational time per solution method and period-window pattern. Every percentage value in the Table and the Figure represents the time increase or decrease of the relevant solution method compared to the classical solution approach. In Table 4.3, the computational time of the classical approach is also presented per pattern. Again, the instances of all problem sets (R1, C1, RC1) have been considered jointly.

Table 4.3: Comparison of computational time of strategies (% difference from the FULL method)

Pattern	FULL (sec)	Difference (%)		
		CLONE	UNI	PARA
1	122	-3%	120%	12%
2	647	-13%	106%	-22%
3	1.495	-27%	75%	-27%
4	2.253	-23%	28%	-27%
5	2.829	21%	44%	-24%
6	7.938	-29%	-27%	-34%
7	12.167	-52%	-58%	-37%
8	12.285	-53%	-60%	-38%
9	14.725	-67%	-74%	-36%
<b>Total/Average</b>	<b>54.460</b>	<b>-46%</b>	<b>-44%</b>	<b>-35%</b>

Considering the computational times over all patterns, the Cloning method seems more efficient with 46% savings compared to 35% savings of the Parallel method. Although the Unified method seems efficient enough (44% reduction on average), it presents the most diverse behavior regarding the period window patterns: It presents the least efficient results for narrow period windows (with even 2 times greater computational time for pattern 1 compared to the classical method); however, for the larger period window patterns it outperforms significantly all other alternatives, succeeding in a 74% reduction for pattern 9. This behavior can be explained by the fact that in the initial patterns, customer flexibility is restricted. Thus, extended labels are usually associated with a limited number of periods; as a result, a large number of labels have to be processed for the same subproblem, increasing computational complexity.

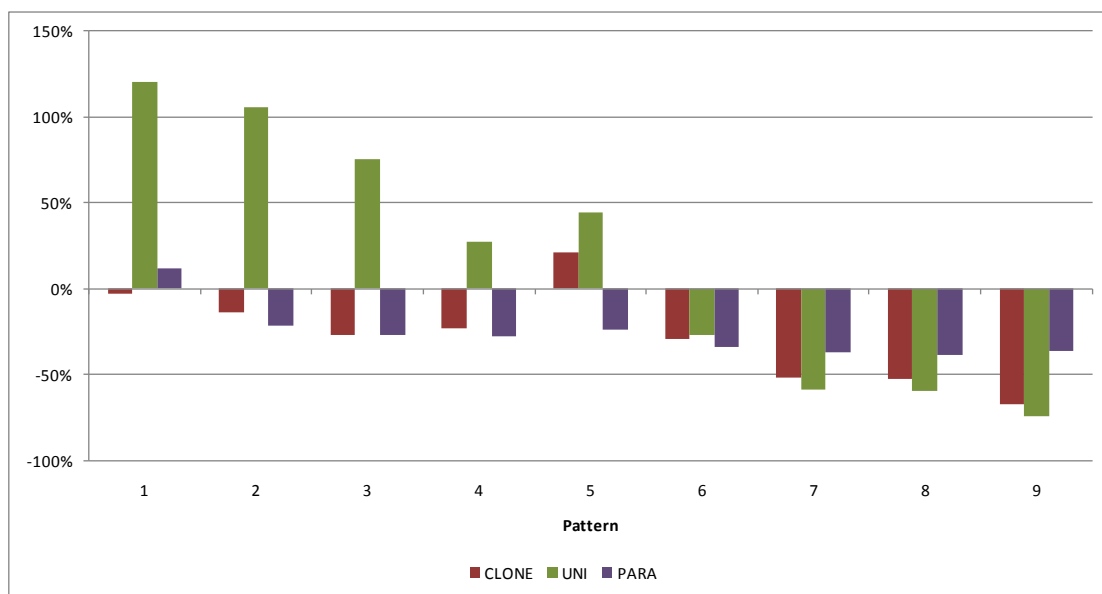


Figure 4.12: Computational time (% difference from the classical method)

## Factor Analysis II: Customer Patterns, Period Window Patterns, Methods

Table 4.4 presents (a) the number of the solved instances over all the multi-period instances per problem sets, and (b) the cumulative computational times per problem set and solution method. Fig. 4.13 illustrates these results. All three alternative methods succeed in reducing the cumulative time across instances of the same problem set, compared to the classical solution procedure. The Cloning method presents the highest overall reduction in the cumulative computational time (46%) in comparison to the classical approach. For the R1 and C1 instances, the cloning method remains the best alternative with time savings of 49% and 48% respectively. For the RC1 instances, the Unified method appears to be the most efficient, resulting to the highest time savings (49%).

Table 4.4: Computational times per problem set (hrs)

Problem Set	Solved Instances	FULL	CLONE	UNI	PARA
R1	105	7,76	3,99	4,22	5,36
C1	73	1,47	0,77	1,29	0,85
RC1	71	5,90	3,34	3,04	3,64
<b>Total</b>	<b>249</b>	<b>15,13</b>	<b>8,10</b>	<b>8,55</b>	<b>9,85</b>

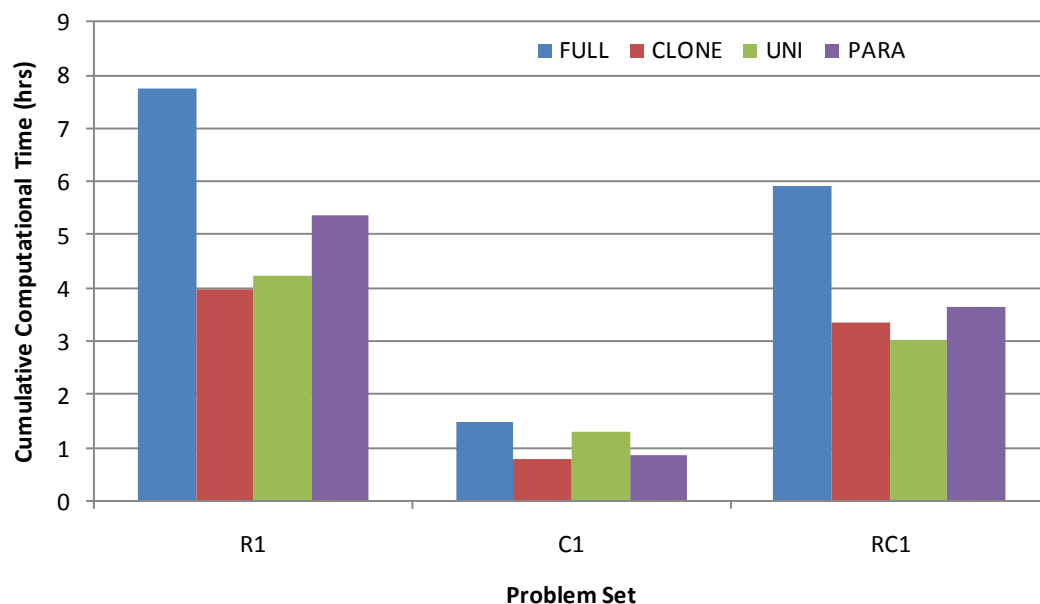


Figure 4.13: Cumulative computational times per problem set and method (hrs)

Figure 4.14 presents the normalized computational times per problem set (e.g. R1, C1 and RC1). The computational times have been normalized with respect to the classical solution

method (100%). As discussed above, the Cloning method provides the largest time savings consistently, although the Unified method is slightly better in the RC1 instances.

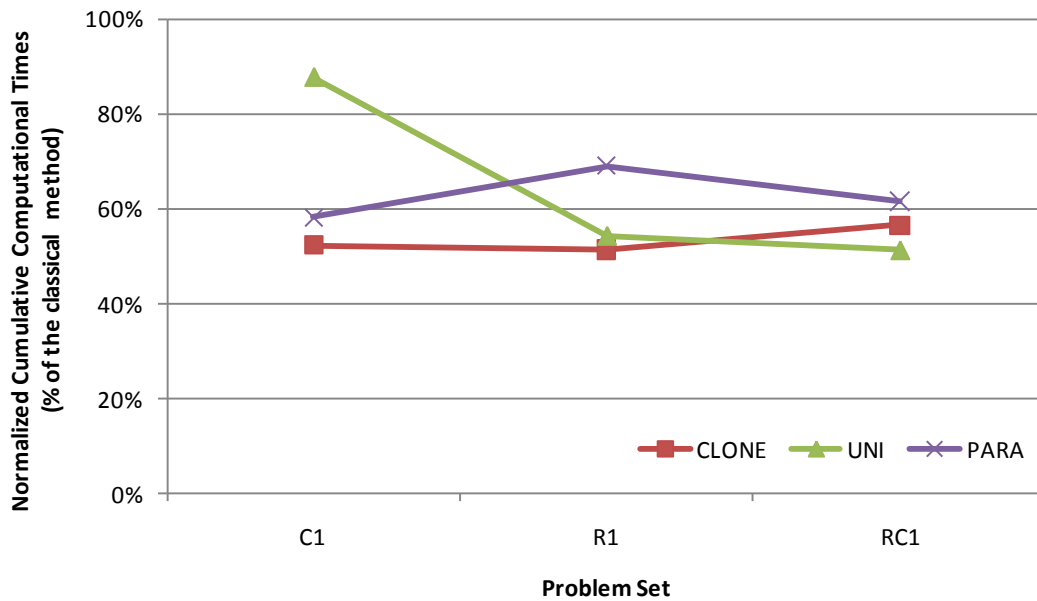


Figure 4.14: Normalized cumulative computational times of alternative methods with respect to the classical approach (100%)

The time savings of all three alternative methods are further analyzed below, based on Figs. 4.15 to 4.17. Again, the computational times of the proposed alternative methods have been normalized with respect to those of the classical solution procedure (100% for each pattern).

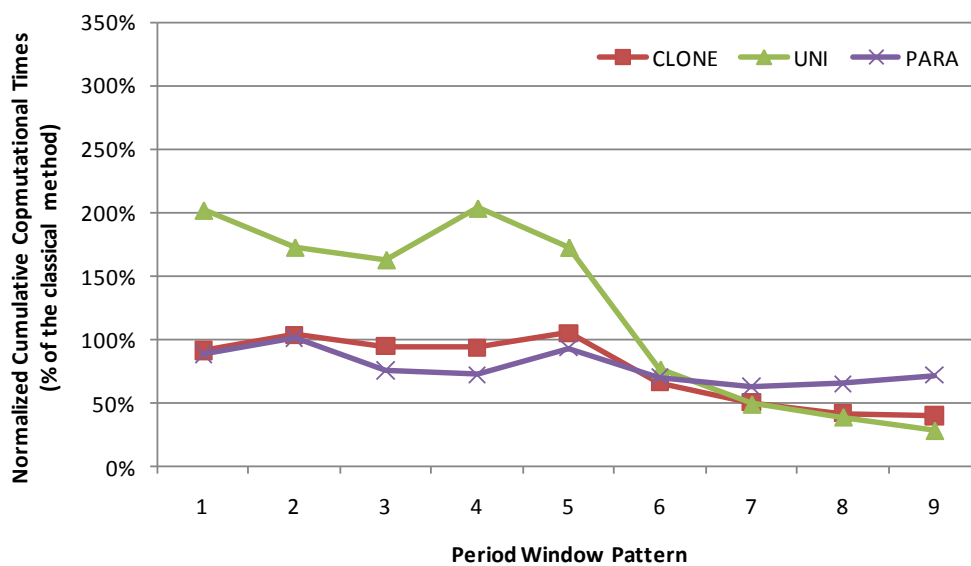


Figure 4.15: Cumulative computational times per pattern for the R1 Instances

In the R1 instances (Fig. 4.15), the Parallel and the Cloning methods appear most efficient. The Parallel method is more efficient in the narrow period windows, while the Cloning method outperforms the former in wide period windows. Furthermore, the Unified method outperforms all other alternative methods for the final three patterns with wider period windows.

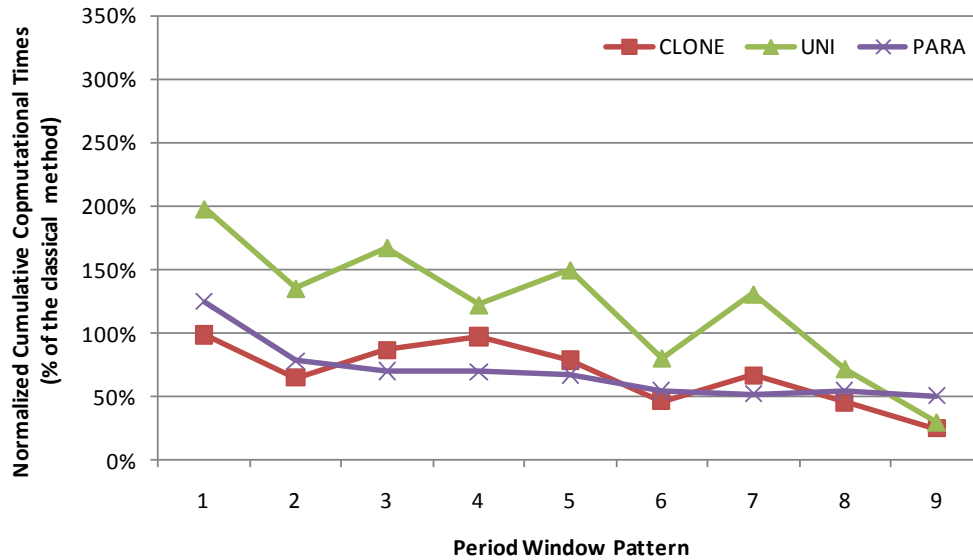


Figure 4.16: Cumulative computational times per pattern for the C1 Instances

For the C1 instances (Fig. 4.16), the Parallel method appears to be the most efficient. Although the Cloning method outperforms the classical approach, its behavior is not as consistent (in comparison to the Parallel one). The Unified method presents the least efficient results among the three methods; even the classical method outperforms the Unified method for the majority of the period window patterns.

For the RC1 instances (Fig. 4.17), the Parallel method presents the most consistent behavior compared to the classical method. On the other hand, the Cloning method, while for some narrow period-window patterns, outperforms the Parallel method, it presents the least efficient results for the moderate period window instances (pattern 5 and 6). Again, the Unified method presents the least efficient results among all solution methods (including the classical approach) for narrow period windows, while outperforms all methods for wider period windows.

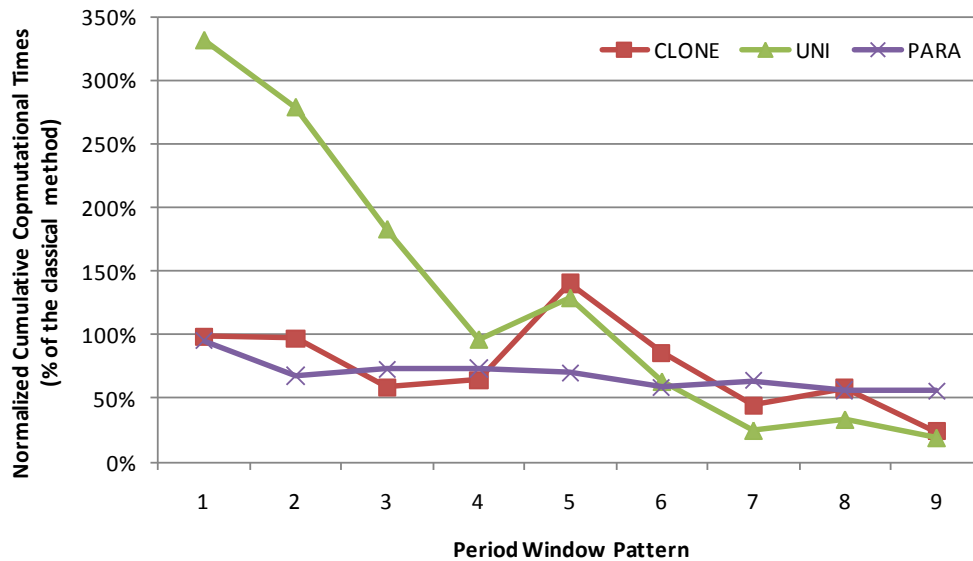


Figure 4.17: Cumulative computational times per pattern for the RC1 Instances

### Summary of conclusions from the analysis

Table 4.5 summarizes the results obtained by the analysis. The Table presents the best alternative method with respect to the various problem attributes (geographical distribution of customers, time windows interval and period window pattern). Period window patterns were categorized in three sets: Narrow (patterns 1 to 3), medium (patterns 4 to 6) and wide (patterns 7 to 9).

Table 4.5: Comparison of alternative methods per factor

		Method		
		CLONE	UNI	PARA
<b>Problem Set</b>	<b>R1</b>	✓		
	<b>C1</b>	✓		
	<b>RC1</b>		✓	
<b>Time Window Category</b>	<b>Narrow</b>	✓		
	<b>Medium</b>	✓		
	<b>Wide</b>		✓	
<b>Period Window Pattern</b>	<b>Narrow (1 to 3)</b>			✓
	<b>Medium (4 to 6)</b>			✓
	<b>Wide (7 to 9)</b>		✓	

The Table indicates that the Cloning method is the most efficient in uniformly distributed (R1) and clustered (C1) problem sets, and in small and medium time windows. The Unified approach performs better in the mixed customer distribution instances (RC1), which are the hardest instances to be solved, and in instances with wide time windows and period windows. Finally, the Parallel implementation of the classical approach outperforms the other

alternatives in narrow and medium period windows, where the customer flexibility is limited. More specifically:

- The Cloning method:
  - Appears to be the most efficient method as far as the total cumulative computational time is concerned (see Table 4.3).
  - Is efficient in R1 and C1 sets with narrow and medium period windows, respectively. Although, specifically in C1 sets, it presents an inconsistent behavior as far as the different period window patterns are concerned ( see Fig. 4.16)
  - In RC1 sets the method is outperformed by the Unified method and also presents an inconsistent behavior regarding the period window patterns (Fig. 4.17).
  - Finally, the Cloning method appears to be efficient in wide period windows regardless the geographical distribution of the customers, and ranks among the two most efficient methods (along with the Unified method).
- The Unified method:
  - Presents the least efficient results in narrow period windows for all three problems sets.
  - In patterns with wide period windows, and especially in R1 and RC1 problem sets, the Unified method outperforms all other methods.
  - Specifically, in C1 instances, it is the least efficient among all methods, including the classical approach, regardless the period window pattern.
- The Parallel method:
  - Appears to be efficient in the narrow period windows, in which the most routes are different per period. Solving all subproblems simultaneously speeds up significantly the classical approach.
  - In patterns with wide period windows, it is less efficient and is outperformed by the other alternative methods. This is because common routes are exploited and duplicate ones are created. Note also that the Parallel method follows similar behavior (although more efficient) to the classical approach in all three problem sets.

## Chapter 5:      **BRANCH AND PRICE: OBTAINING INTEGER SOLUTIONS**

In order to obtain integer solutions for the Multi-Period Vehicle Routing Problem with Time Windows (MPVRPTW), we embedded the Column Generation approaches described in Chapters 3 and 4 within a branch-and-price (B&P) scheme. B&P divides the feasible solution space into subspaces, by avoiding selected fractional values of the problem variables (Lawler and Wood, 1966; Lee and Mitchell, 2001). The solution space is repeatedly divided until integer solutions are obtained, in a similar way to the classical branch-and-bound (B&B) procedure.

Additionally B&P allows the generation of new proposed columns, i.e. routes, by solving the subproblems at each node of the B&P tree. That is the fundamental distinction between B&P and the classical branch-and-bound (B&B) procedure. Thus, the term "price" refers to the pricing procedure that generates new columns (routes).

In this Chapter, we first discuss existing B&P methods initially proposed for the VRPTW (Section 5.1). Section 5.2 proposes ways to adapt the existing VRPTW B&P methods to the MPVRPTW. Section 5.3 proposes a heuristic technique to explore the B&P tree in an efficient manner and obtain near optimal solutions with significant computational savings. Finally, in Section 5.4 we test all aforementioned methods using the testbed developed and presented in Chapter 4.

### 5.1 THE GENERIC BRANCH AND PRICE FOR THE VRPTW

Figure 5.1 describes the generic B&P procedure. Following the classic B&B procedure, B&P procedure starts with obtaining the overall Lower Bound ( $LB$ ), as described in Chapter 3 and 4. At this initial stage, the Global Upper Bound (GUB), i.e. the best known integer solution, is set equal to a large number ( $M$ ) or equal to the cost of any initial feasible integer solution (e.g. obtained by a heuristic). If the  $LB$  corresponds to an integer solution, the algorithm terminates and the  $LB$  is the optimal integer solution. In case the solution is not integer, the *Branching Policy* is triggered. Given a fractional solution, the *Branching Policy* divides the feasible solution space into subspaces, by avoiding fractional values of the selected problem variables. Each subspace is explored (i.e. solved using the column generation method) independently

and can lead to additional solution space division, in cases where fractional solutions are obtained. Each one of these subspaces represents a new node of the B&P tree structure. Each B&P node that is explored is deleted and the new nodes that are created are added to a list of the remaining unexplored nodes (*BBlist*). In case a better integer solution (than the integer solution found so far) the GUB is updated to these integer solution. Continuing the method, another policy (*Node Selection*) is triggered in order to select the next node, among the available unexplored nodes, to be explored. The algorithm terminates when no more unexplored nodes exist or when the LB of all remaining nodes is larger than the GUB (pruning of B&P nodes). Note that every B&P node is characterized by its predecessor (father node) LB. Based on that, a node with LB larger than the current GUB cannot further improve the solution, since its solution will always be larger than the predecessors' node LB.

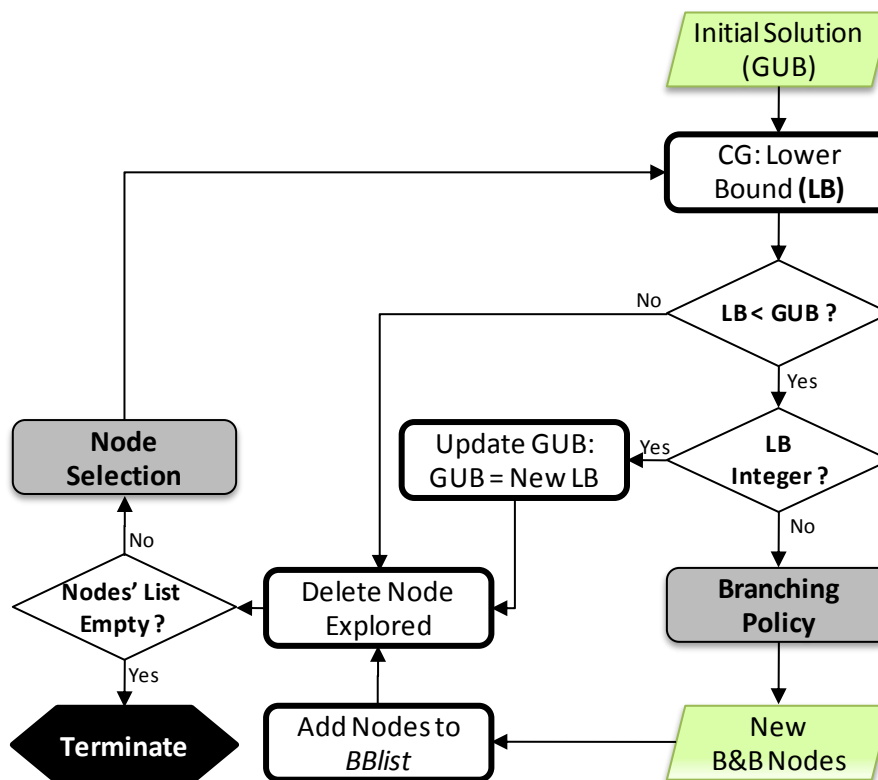


Figure 5.1: Branch and Price procedure

### Branching Policies

Branching policy is used when a fractional LB solution is found for the current B&P node. The policy is used to select the variable to partition the solution space. Some commonly used branching policies on the VRPTW are overviewed below (Larsen, 2001; Danna and Le Pape, 2005):

### Branching on the Number of Vehicles

This was proposed initially by Desrochers *et al.* (1992). Given a solution with a fractional number of vehicles equal to  $\mathcal{F}$ , the solution space is divided into two subspaces.

Let  $\mathbf{X}_{ij} = \sum_{k \in V} x_{ijk}$  be the cumulative vehicle flow from arc  $(i, j)$  in the final fractional solution; that is the number of vehicles passing through arc  $(i, j)$ .  $\mathbf{X}_{ij}$  provided by the routes in the final fractional solution of the RMP. Based on  $\mathbf{X}_{ij}$ , two independent subspaces are created. The first is defined by an additional constraint,  $\sum_{j \in N} \mathbf{X}_{0j} \geq \lceil \mathcal{F} \rceil$ , while the second by  $\sum_{j \in N} \mathbf{X}_{0j} \leq \lfloor \mathcal{F} \rfloor$ . Note that these constraints can be easily incorporated in the RMP formulation, and define two new nodes.

Although an integer number of vehicles may be obtained, using this policy, the actual LB solution may still be fractional, with respect to the flow variables. For that reason, this strategy is generally used together with other strategies.

### Branching on Flow Variables

Given a solution with a fractional variable  $x_{ijk}$ , the first subspace is defined by the additional constraint,  $x_{ijk} = 1$ , while the second is defined by,  $x_{ijk} = 0$ . Thus, the first constraint forces vehicle  $k$  to pass through arc  $(i, j)$ , while the latter forbids it.

Since in the classical VRPTW, vehicles are considered to be identical, this branching cannot be incorporated in the B&P procedure easily. That is, because branching on a specific vehicle  $k$  is not possible since all vehicles are dealt through the same common subproblem and the vehicles are not independently identified in the master problem. In order to incorporate it, one subproblem per vehicle needs be solved. Due to this reason, branching on single flow variables is not widely used.

### Branching on Sums of the Flow Variables

This branching strategy was proposed by Halse (1992) and Desrochers *et al.* (1992). Given a fractional solution of a B&P node, the sum of the vehicle flows ( $\mathbf{X}_{ij} = \sum_{k \in V} x_{ijk}$ ) is calculated. Selecting a fractional  $\mathbf{X}_{ij}$ , the first subspace is defined by constraining arc  $(i, j)$  to be excluded by the solution, while the other subspace ( $\mathbf{X}_{ij} = 1$ ) constrains arc  $(i, j)$  to be part of the solution.

A major advantage of this strategy is that it can be easily implemented, without adding the new constraints explicitly to the RMP formulation, but instead by modifying the cost matrix

of the subproblem. Considering the first subspace, (a) all routes containing customers  $i$  and  $j$  not in a consecutive manner are discarded from the current RMP of the father node, and (b) all cost coefficients  $c_{il}, \forall l \neq j$  and  $c_{lj}, \forall l \neq i$ , are set to  $\infty$ . These two modifications will allow connections only from customer  $i$  to customer  $j$  to be created. In the second subspace, the routes containing the arc  $(i, j)$  are discarded, and only the coefficient  $c_{ij}$  is set to  $\infty$ .

### Selection of Branching Variable

Regardless of the branching policy, a decision needs to be taken regarding the most promising variable to branch on. Simple heuristics are typically used in order to select the branching variable among the set of fractional variables. The most common approach is to branch on the variable with the most fractional value (that is the value  $f$ , for which  $([f] - f)$  is closest to 0.5).

### **Node Selection Policy**

The node selection policy consists of the method to search and solve the known nodes of the branch-and-price tree, that is the set of the known fractional solutions. There are several policies, the majority of which mimics tree search methods, such as depth-first, best-first, width-first and depth first with backtracking (see Larsen, 2001; Lee and Mitchell, 2001). One of the most widely-used approaches is the Best-First approach. In a set of unexplored branch-and-bound nodes, every node has been assigned with a metric, which is either the lower bound of the CG solution corresponding to the parent node, or the lower bound corresponding to the node itself. In the latter case, tighter bounds are assigned to the nodes and if a good GUB value exists, many nodes may be discarded prior to expansion. On the other hand, calculation of these lower bounds consumes considerable computational time. Based on this metric, the node with the lowest LB is selected to be explored next.

## **5.2 BRANCH AND PRICE FOR THE MULTI-PERIOD ROUTING PROBLEM**

### **Branching techniques**

The branching and node selection policies proposed for the VRPTW (Section 5.1) are modified properly in order to manage the additional degree-of-freedom introduced by the flexibility of serving customers in multiple, adjacent, periods. Note that, as a consequence, all decision variables ( $x_{ijkp}$ ) of the original formulation of the problem, contain an additional subscript ( $p$ ) denoting the associated period to the decision variable.

One of the most widely used branching methods for the VRPTW is the "branching on the sum of the flow variables over vehicles". This method is used either on its own, or together with other branching methods (see Section 5.1). In the remainder we discuss our adaptation of the above method for solving the MPVRPTW.

Consider a fractional lower bound and let  $X_{ijp} = \sum_{k \in K^p} x_{ijkp}$  be the cumulative vehicle flow of arc  $(i, j)$  in period  $p$ , and  $X_{ij} = \sum_{p \in P} X_{ijp}$  be the cumulative vehicle flow of arc  $(i, j)$  over all periods. We select to branch on the most fractional  $X_{ijp}$ . In order to implement this, one can select either of two branching methods for the B&P tree (see Fig. 5.1):

- *2br method*: Two B&P nodes are created: The first concerns the subspace defined by the additional constraint  $X_{ijp} = 0$ , and does not allow arc  $(i, j)$  to exist within period  $p$  in any of the solutions within the defined subspace; however, this arc is allowed to exist in all other periods. The second B&P node is defined by  $X_{ijp} = 1$ , which forces arc  $(i, j)$  to be part of the solution within period  $p$ .
- *$P + 1$  method*: This method creates at least 2 B&P nodes. The actual number of nodes to be created depends on the branching arc  $(i, j)$  and the feasibility of customers  $i$  and  $j$  to be routed within the said periods. The first  $|P|$  B&P nodes are defined by  $X_{ijp} = 1$ , one per each period  $p$ , respectively. Each node forces arc  $(i, j)$  to be part of the solution within the relevant period  $p$ . Note that in cases in which customers  $i$  and  $j$  are not both feasible within a period, arc  $(i, j)$  is also not feasible. Thus, the related subspace is not explored, and, therefore, the B&P node that corresponds to this period is not created. An additional B&P node is generated for the subspace in which  $X_{ij} = 0$ , and arc  $(i, j)$  does not participate in any solution. In this B&P node, arc  $(i, j)$  is excluded from all periods of the planning horizon.

Figure 5.1 illustrates the two branching methods using a sample problem with 3 periods. In this example, it is assumed that customers  $i$  and  $j$  are feasible in all three periods, and  $(i, j)$  is the only arc with fractional flow within period  $p = 1$ . Given the initial lower bound (LB), the *2br* method will divide the solution space three times, resulting in 6 subspaces; that is,  $X_{ij1} = 1$  and  $X_{ij1} = 0$ , with the other subspaces created in a similar way in case fractional solutions with respect to arc  $(i, j)$  continue to appear within period 2 ( $X_{ij2} = 1$  and  $X_{ij2} = 0$ ) and 3 ( $X_{ij3} = 1$  and  $X_{ij3} = 0$ ). On the other hand, the  *$P + 1$*  method will result directly in 4 subspaces. Three subspaces are defined by  $X_{ijp} = 1, p = 1, 2, 3$ , and one is defined by  $X_{ij} = 0$

(i.e.  $X_{ij1} + X_{ij2} + X_{ij3} = 0$ ). Note that in Fig. 5.2, the highlighted subspaces are identical, since  $X_{ij} = \sum_{p \in P} (X_{ijp})$ .

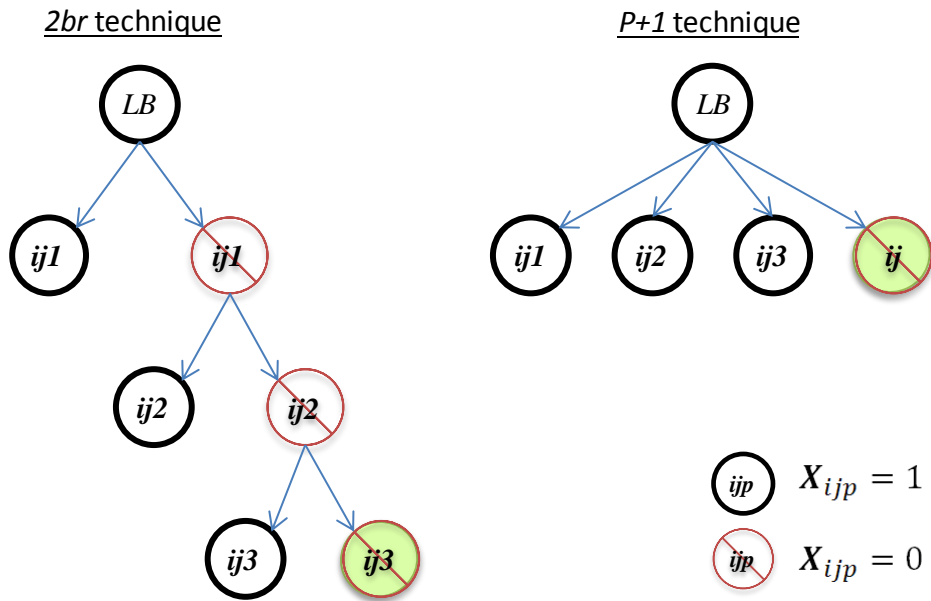


Figure 5.2: Multi period Branch and Price techniques

The motivation of examining both methods is as follows: Considering a full exploration of both trees, the  $P + 1$  method is expected to be more efficient than the  $2br$  method, since the former does not explore certain subspaces (e.g.  $X_{ij1} = 0$  and  $X_{i2p} = 0$ ). For example, in the  $2br$  method and for instances with wide period windows, the branch tree node  $X_{ijp} = 0$  does not define a "strong" partition of the solution space, since fractional variables are still allowed for periods other than period  $p$ . That is, a fractional solution which we branched upon may be replicated by traversing arc  $(i, j)$  within other periods, resulting to similar fractional solutions.

On the other hand, if an integer solution has been found at one of the nodes of the  $2br$  method, its successors will not be created. For example, if an integer solution is obtained in the subspace defined by  $X_{ij1} = 0$ , the four successor subspaces will not be created by the  $2br$  method. In this case,  $2br$  is expected to be faster.

### Implementation Issues

Both branching methods can be implemented without modifying/adding additional constraints to the RMP, or the subproblems. This can be achieved by: (a) Modifying the time matrix of each period appropriately in order to avoid creating new routes that contain the "forbidden" arcs (note that for each period a separate time and cost from/to matrix is maintained, see

Section 3.2), and (b) removing the existing routes within the RMP that violate the additional constraints of the relevant B&P node.

Let  $t_{ijp}$  be the travel time of traversing arc  $(i, j)$  in period  $p$ ,  $N_p$  be the set of feasible customers within period  $p$ , and  $H$  be the planning horizon (periods  $1, \dots, P$ ).

- Considering subspace ( $X_{ijp} = 0$ ), (a) all routes within the RMP that is relevant to period  $p$  and contain arc  $(i, j)$  are discarded (removed from the RMP of the said B&P node), and (b) the coefficient  $t_{ijp}$  is set to  $\infty$ . Thus, traversing arc  $(i, j)$  is not allowed in period  $p$ .
- Considering subspace ( $X_{ijp} = 1$ ), we disregard (remove from the RMP of the said B&P node) the following (a) all routes within period  $p$  that contain either customer  $(i$  or  $j)$ , or both customers not in a consecutive manner (i.e. arc  $(i, j)$  does not exist), and (b) all routes of the other periods that contain either customer  $i$  or  $j$ . Additionally, all travel time coefficients  $t_{ilp'}$  and  $t_{ljp'}$ ,  $\forall l \in N_{p'}, \forall p' \in H$  are set to  $\infty$ , except from the coefficient  $t_{ijp}$  of period  $p$ . Thus, traversing arc  $(i, j)$  will always be part of the solution within period  $p$ .
- Especially in the  $P + 1$  method, and for the subspace defined by constraint  $X_{ij} = 0$ , all routes containing arc  $(i, j)$  within all periods are discarded, and the coefficients  $t_{ijp}$  for every period  $p$  are set to  $\infty$ . Thus, traversing arc  $(i, j)$  is not allowed in any period.

#### The special case of Unreachable Nodes in the ESPPTWCC

In our implementation, we utilize the concept of unreachable nodes proposed by Feillet *et al.* (2005). In this concept, every label contains a binary vector with one element per customer. The unreachable nodes, i.e. the already visited nodes, as well as the "non-feasible to be visited" nodes due to constraint limitations, are set equal to 1 within this vector. Thus, a label can be extended only to nodes that are not included in its related set of unreachable nodes.

Note, however, when branching is implemented with the aforementioned modifications (i.e. setting values  $t_{ijp}$  to  $\infty$ ), the triangular inequality may no longer be valid. For example, consider a case with 3 customers in which the only fractional solution is related to arc  $(2, 3)$  and, thus, branching on this arc is performed. Note that the period subscript has been dropped, since it is not relevant in this example. Considering the subspace defined by the additional constraint  $x_{23} = 1$ , all  $t_{2j}, \forall j \neq 3$  are set to  $\infty$  while  $t_{23}$  remains as is. Given partial path  $\delta = [1 \ 2]$ , the associated unreachable vector  $R_{2\delta}$  is equal to  $[1 \ 0 \ 1]$ . (Each unreachable vector

$R_{i\delta}$  contains all customers excluding the depot and thus  $R_{2\delta} = [r_{2\delta}^2, r_{2\delta}^3, r_{2\delta}^4]$ , where  $r_{2\delta}^i = 1$  if customer  $i$  either has already been visited or cannot be visited due to resource limitations). That is, path  $\delta$  will be extended to node 3 only, and a new partial path  $\delta' = [1\ 2\ 3]$  will be defined.

Up to this point the whole procedure has been implemented properly; that is, arc (2,3) is part of the generated paths. However, since the associated label  $L_{3\delta'}$  inherits the unreachable vector from its predecessor, customer 4 cannot be visited (note that  $r_{2\delta}^4 = 1$  and, thus,  $r_{3\delta'}^4 = 1$ ). In such case, partial path  $\delta'$  will be prohibited to be extended to other remaining nodes (e.g. node 4); this, of course, precludes feasible solution regions and may lead to suboptimality.

In order to address this situation, the time matrices are not modified. Instead, we store the information of the branching decisions (i.e. which arcs to prohibit from traversing in certain periods) in a separate binary matrix  $\bar{B}_p$  for each period  $p$ , with elements:

$$\bar{b}_{ijp} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is allowed in period } p \\ 0 & \text{otherwise} \end{cases} \quad (5.1)$$

Thus, when solving a subproblem, a label  $L_{i\delta}$  is allowed to be extended to another label  $L_{j\delta'}$  only if the relevant  $\bar{b}_{ijp}$  equals to one; otherwise the extension is not performed.

### 5.3 HEURISTIC PRUNING FOR BRANCH AND PRICE

Although B&P is not exhaustive (but exact), it may be excessively expensive, or, even, computationally intractable. For example, even in cases in which the optimal solution has been obtained, in order to prove optimality the method still needs to solve a subset of the remaining B&P nodes (e.g. those with lower bound lower than the integer solution) (see Section 5.3).

To obtain efficient integer solutions faster, we propose a heuristic pruning technique which discards non-promising nodes (subspaces) of the B&P tree. The goal is to solve less B&P nodes, while obtaining efficient integer solutions. Note that for these solutions optimality cannot be proven.

Recall that when solving a node ( $n$ ) of the B&P tree we obtain a lower bound  $LB_n$ . Additionally, an upper bound of the integer solution of node  $n$  may be calculated (denoted as  $IB_n$ ) by solving a Branch and Bound problem, using only the columns (routes) that exist in

the current RMP ( $\cup_{p \in P} \Omega'_p$ ). The latter may be calculated using the default integer programming methods of the CPLEX environment. Given the current best known integer solution up to that point (i.e. Integer Upper Bound  $IUB$ ), the following metric for each node ( $n$ ) may be calculated:

$$M_n = \frac{IUB - LB_n}{IB_n - LB_n} \quad (5.2)$$

If  $LB_n$  is larger or equal to  $IUB$  (i.e.  $M_n \leq 0$ ), the corresponding B&P node is discarded by the fathoming rules of the classic B&P procedure. That is, an improvement on  $IUB$  is not feasible, and node  $n$  is discarded without creating any child nodes. Note that this procedure is maintained within the proposed heuristic.

If both  $LB_n$  and  $IB_n$  are lower than the current  $IUB$  (i.e.  $M_n \geq 1$ ), then a new better integer solution has been obtained ( $IB_n$ ) and is set as the new  $IUB$ . In this case, node  $n$  will always be further explored, since an improvement on  $IB_n$  is possible.

The proposed heuristic pruning technique is triggered when  $0 < M_n < 1$ . In such cases,  $LB_n < IUB < IB_n$  and the value of  $M_n$  provides an insight on the current quality of the B&P node, i.e. its ability to improve the current best integer solution ( $IUB$ ). Note that in the exact B&P approach, node  $n$  is explored further (not discarded). In our proposed heuristic we select to discard node ( $n$ ) when the potential for improvement of the best integer solution up to that point ( $IUB$ ) appears to be limited. The potential for improvement of  $IUB$  is assessed by the relative distance between  $IUB$  and  $LB_n$  as compared to the distance between  $IB_n$  and  $LB_n$ . The greater the relative distance is, the greater the potential for improvement appears to be. Based on this argument, node  $n$  is discarded when this relative distance is low, or, equivalently,  $M_n$  is lower than a threshold value  $\lambda \in (0,1)$ .

This argument is illustrated in Figure 5.2. Given node ( $n$ ) and its corresponding  $LB_n$  and  $IB_n$ , if the current best integer solution ( $IUB$ ) is relatively closer to  $LB_n$ , then the exploration of node  $n$  is not expected to improve  $IUB$  (the current best integer solution) significantly, since the interval  $(LB_n, IUB)$  is narrow. Additionally, even if there is a better integer solution within this interval, the cost improvement will be limited (Fig. 5.3a). Conversely, if  $IUB$  is close to  $IB_n$  then the exploration of node  $n$  provides a more significant potential for obtaining a better integer solution than the current  $IUB$  (Fig. 5.3b).

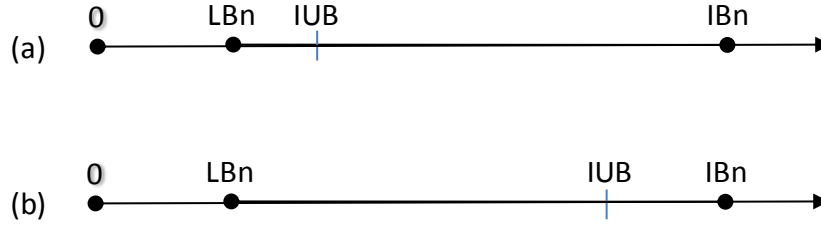


Figure 5.3: Example of the heuristic pruning technique

In the extreme case of  $\lambda = 0$ , all B&P nodes will be explored, resulting in an exact B&P procedure. In the other extreme case ( $\lambda = 1$ ), all nodes are discarded and, thus, Branch and Bound is performed only on the columns obtained by the solution of the global lower bound (Chapter 3.2). The latter case ( $\lambda = 1$ ) has been utilized by Tricoire (2006).

## 5.4 BRANCH AND PRICE TESTING

The parameters used and choices made in our implementation are as follows:

- Exploration of the B&P nodes is performed using a best-first node selection policy, that is, the node with the lowest lower bound is explored next (note that each node is characterized by the lower bound of their parent node)
- All column generation parameters remain the same as in Section 3.3.5
- The Cloning method was utilized for obtaining the lower bound of each instance (except in 5.4.3 where both the classical column generation and the Cloning method are compared)
- When branching on the sum of flow variables and periods, and if more than two arcs are competing for branching, then we select the one that is feasible in less periods; this way, fewer B&P nodes need to be explored
- Each test instance is terminated when the computational time exceeds a time limit (set to 1 hr in our implementation).

In Subsection 5.4.1 we validate the implementation of the B&P method by solving suitably modified Solomon benchmarks, for which the optimal solutions are known. Subsection 5.4.2 presents comparison tests of the proposed B&P methods ( $P + 1$  and  $2br$ ) regarding computational efficiency. Subsection 5.4.3 presents comparison tests of the B&P scheme when using the classical column generation approach and when using the cloning method. Finally, Subsection 5.4.4 analyzes the efficiency of the proposed heuristic pruning technique (Section 5.3) with respect to different values of the threshold  $\lambda$ .

### 5.4.1 VALIDATION OF THE BRANCH AND PRICE IMPLEMENTATION

In order to validate our branch-and-price implementation, we solve appropriate instances from those presented in Chapter 4. Specifically we select the instances of pattern 9, in which the period windows of all customers span the entire planning horizon. In addition, note the number of available vehicles (within all periods) is large enough to satisfy the demand for each instance.

Recall that all instances of Chapter 4 were based on the Solomon benchmarks for single period VRPTW, which were converted to MPVRP instances by applying a period window to each customer. Given that in the selected instances the period windows span the entire planning horizon, the optimal solution of the MPVRP is the same as the one of the VRPTW, which is known for the selected problems (Larsen, 2001). The goal of the validation testing for our MPVRP solution approach is to obtain these optimal solutions in the MPVRP setting.

Table 5.1 presents the problem instance, the lower bound and the relevant computational time, the integer solution obtained, the total B&P nodes created, the nodes explored until the optimal solution is reached, the B&P node in which the first occurrence of the optimal solution was detected and the computational time for the completion of the algorithm. In instances where the "integer solution" fields are empty, the optimal integer solution was obtained directly by the lower bound calculation and, thus, B&P was not needed. For the instances that are not reported, the algorithm either terminated by the time limit or by resource (memory) overflow, and the optimal integer solution was not obtained.

Table 5.1: Validation of the branch and price algorithm using Solomon benchmarks

Problem		Vehicles per Period	Lower Bound		Integer Solution				
#	Set		Cost	Time	Cost	Total Nodes	Nodes Explored	First Occ. <sup>(1)</sup>	Time (sec)
9	r101	4	1043.4	6.14	1044.0	6	6	2	27.47
18	r102	3	909.0	35.63	(2)				
27	r103	3	769.2	34.37	772.9	141	141	1	2553.61
45	r105	3	892.1	10.46	899.3	559	199	86	1379.16
54	r106	3	791.4	26.92	793.0	43	13	13	233.83
62	r107	3	707.3	79.37	711.1	95	46	18	1763.34
90	r110	2	695.1	38.60	697.0	109	43	19	1019.18
117	c101	2	362.4	38.58	(2)				
126	c102	3	361.4	138.72	(2)				
135	c103	2	361.4	742.25	(2)				
153	c105	2	362.4	65.35	(2)				
162	c106	2	362.4	39.87	(2)				

Problem		Vehicles per Period	Lower Bound		Integer Solution				
#	Set		Cost	Time	Cost	Total Nodes	Nodes Explored	First Occ. <sup>(1)</sup>	Time (sec)
171	c107	2	362.4	66.90	(2)				
180	c108	2	362.4	92.11	(2)				
189	c109	2	362.4	215.32	(2)				
225	rc104	2	545.8	1568.86	(2)				
261	rc108	2	541.2	1967.52	598.1	9	5	1	3633.01

(1) Number of nodes explored to initially reach the optimal integer solution.

(2) Optimal solution reached by the Column Generation method directly.

For all instances solved, the solutions obtained for the MPVRPTW were identical to the optimal integer solutions reported in the literature for the VRPTW. This provides a strong indication for the validity of our B&P (and CG) implementation.

#### 5.4.2 COMPARISON OF BRANCH AND PRICE TECHNIQUES ( $P + 1$ AND $2br$ )

We tested both proposed branching techniques ( $P + 1$  and  $2br$ ) described in Section 5.1 and compared the results obtained in terms of computational efficiency, using the 66 instances that required the use of B&P and were solved by both techniques.

Appendix B.1 presents detailed results of the two  $B\&P$  methods regarding these 66 instances. Table 5.2 summarizes these results presenting the average computational time per instance, the average number of nodes explored, and the average number of nodes explored until the optimum was initially reached (first occurrence of optimal solution).

Table 5.2: Aggregate B&P results for the 66 instances solved by both techniques ( $2br$  and  $P + 1$ )

Method	Average Time (sec)	Nodes Explored	First Occ. <sup>(1)</sup>
$P + 1$	384.8	44.5	27.1
$2br$	391.8	46.5	21.0

(1) Number of nodes explored since the optimal integer solution was initially reached

It is clear from these aggregate results, as well as the detailed ones in Appendix B.1, that the average performance of the two techniques with respect to computational times and number of nodes explored is almost identical. Note that the  $2br$  method appears to converge to the optimal integer solution by solving fewer nodes, as evidenced by the reduction in the number of nodes explored until the optimal integer solution was initially reached. However, no method outperforms significantly the other one, in the average sense.

### 5.4.3 COMPARING FULL AND CLONE METHODS WITH B&P

Based on the results of Chapter 4, for the B&P implementation we selected

- the CLONE over the UNI method, since it appears to: (a) yield the most efficient cumulative computational times and (b) be the most consistent across all patterns and client geographical distributions.
- The *2br* branching technique.

We compared the efficiency of the B&P scheme that uses the CLONE method versus the B&P scheme that uses the reference method (FULL).

Table 5.3 presents the number of test instances that converged either to the optimal integer solution, or obtained an integer solution within the time limit (one hour of computational time). The first column presents the instances for which the optimal integer solution was obtained directly by solving the relaxed problem with the column generation method without the use of B&P. The second column presents the instances that converged to the optimal integer solution using B&P, while the third column presents the instances that converged to an integer solution but terminated due to the time limit. The remaining instances did not obtain an integer solution within the one hour time limit and were not tested further. Note that most of these latter instances belonged to the RC1 configuration.

Table 5.3: Instances with an integer solution using the FULL or CLONE method

Method	Integer by CG	Integer by B&P	Integer Solution (optimality not verified)	Total
FULL	89	70	37	196
CLONE	92	72	37	201

Figure 5.4 analyzes the average ratio of the B&P computational times using the CLONE method versus the FULL method over all client geographical distributions per period window pattern. Two different ratios are presented: (a) The IB ratio that concerns all 196 instances for which an integer solution was obtained by both methods, and (b) The IC ratio that concerns only the instances which were solved to optimality by B&P.

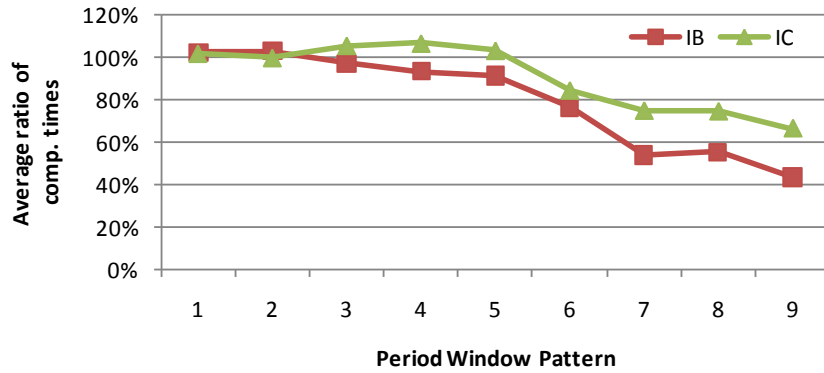


Figure 5.4: Average computational time ratio (CLONE vs. FULL)

The IB ratio curve shows that the CLONE method results in significant gains in determining the optimal (or a suboptimal) integer solution, especially as the width of the period window increases. The IC ratio curve indicates that the efficiency gains of the CLONE method are moderated when it is used in the B&P scheme. This is attributed to the fact that the savings, stemming from determining the lower bound, are moderated by the other B&P operations, such as the generation of the B&P nodes.

Figure 5.5 presents the average ratio of the number of B&P nodes explored by the CLONE versus the FULL method per each period window pattern and across all client distributions. Two different ratios are presented: The first concerns all instances for which both methods determined the optimal integer solution within the time limit using B&P. The second concerns those instances for which an integer solution was reached but the B&P scheme was terminated due to the time limit. In the latter case, since CLONE is more efficient, it is expected to explore a greater number of nodes within the same time interval. This is verified by the second ratio in Fig. 5.5. This also seems to be the reason that five more instances were solved by the B&P using the CLONE method vs. the FULL (see Table 5.3). The first ratio verifies that in reaching the optimal solution, approximately the same number of nodes is explored by either B&P method.

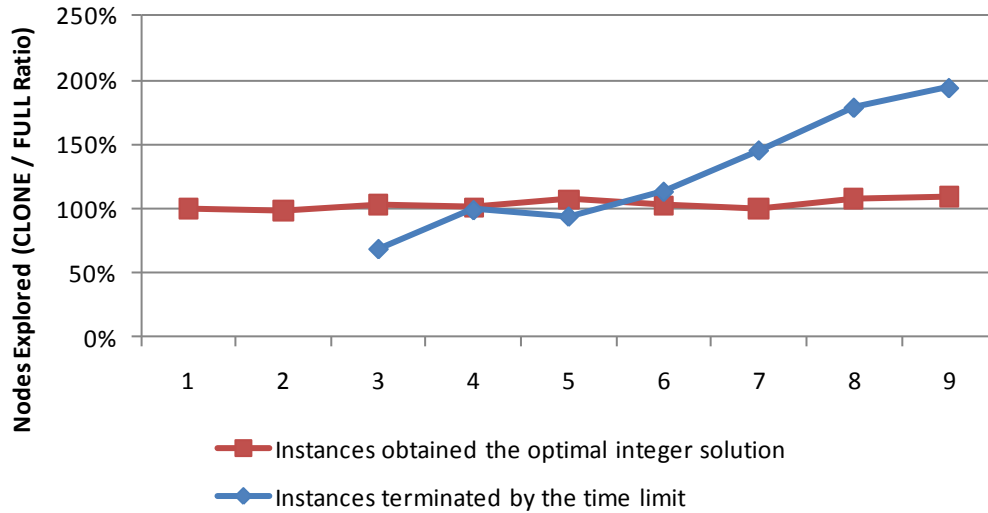


Figure 5.5: Nodes explored per period window pattern

#### 5.4.4 TESTING OF PROPOSED HEURISTIC PRUNING TECHNIQUE

In order to analyze the proposed heuristic technique described on Section 5.3, different values for the threshold ( $\lambda$ ) were tested. Recall that given a value for the threshold  $\lambda \in [0,1]$ , the B&P nodes for which  $M_n > \lambda$  are maintained and explored further. The rest are discarded.

The analysis was based on the set of 66 problem instances for which the optimal solutions were obtained by both *B&P* techniques. Analytical results are reported in Appendix B.2. Tables 5.4 and 5.5 present the aggregate results of the analysis for different values of ( $\lambda$ ) using the  $P + 1$  and  $2br$  techniques, respectively. Specifically the two Tables present the average percent cost difference with the optimal integer solution, the average number of nodes explored per problem, the average number of nodes explored until the best integer solution was initially reached, the average computational time per problem, and the number of instances for which the optimum was not obtained. (Note that the cost difference statistic in Tables 5.4 and 5.5 considers only the instances for which the optimal integer solution was not obtained. The other statistics consider all instances).

Table 5.4: Performance of pruning heuristic for different values of threshold  $\lambda$  using the  $P + 1$  branching method (66 instances solved)

$\lambda$	Instances in which optimal was not reached	Cost Difference	Nodes Explored		Average Comp. Time per Instance (sec)
			Average	First Occ. <sup>(1)</sup>	
<b>1.00</b>	25	0.456%	1.0	1.0	23.2
<b>0.99</b>	6	0.389%	10.6	3.7	163.8
<b>0.95</b>	5	0.354%	12.1	3.6	169.3

$\lambda$	Instances in which optimal was not reached	Cost Difference	Nodes Explored		Average Comp. Time per Instance (sec)
			Average	First Occ. <sup>(1)</sup>	
<b>0.85</b>	4	0.182%	14.2	3.7	176.9
<b>0.75</b>	3	0.193%	17.3	4.1	191.0
<b>0.50</b>	-	-	22.5	5.4	226.4
<b>0.00</b>	-	-	44.5	27.1	384.8

(1) average number of nodes explored since the optimal integer solution was obtained.

Table 5.5: Performance of pruning heuristic for different values of threshold  $\lambda$  using the **2br** branching method (66 instances solved)

$\lambda$	Instances in which optimal was not reached	Cost Difference	Nodes Explored		Average Comp. Time per Instance (sec)
			Average	First Occ. <sup>(1)</sup>	
<b>1.00</b>	25	0.456%	1.0	1.0	22.8
<b>0.99</b>	5	0.233%	10.0	3.6	142.4
<b>0.95</b>	5	0.233%	10.6	3.7	145.7
<b>0.85</b>	5	0.233%	14.3	4.4	171.2
<b>0.75</b>	4	0.151%	17.1	5.4	187.9
<b>0.50</b>	1	0.351%	22.3	6.8	215.0
<b>0.00</b>	-	-	46.5	21.0	391.8

(1) average number of nodes explored since the optimal integer solution was obtained.

From Tables 5.4 and 5.5 it is clear that the proposed heuristic method presents similar results under both the 2br and the P+1 B&P techniques.

As far as the solution quality is concerned, the heuristic results in very limited deviations from the optimal integer solutions. Even with  $\lambda = 1$ , only in 25 out of 66 instances the optimal solution was not reached. The maximum cost deviation over all instances in both cases was equal to 2.07%. Note that even for  $\lambda = 0.99$  the average deviation from the optimal integer solutions is very limited and equal to 0.233%.

In terms of computational time, the proposed heuristic results in significant computational time savings. For example, for  $\lambda = 1$  a good solution is reached within 6% of the time required to obtain the optimal solution (that is, for  $\lambda = 0$ ). Even for  $\lambda = 0.5$ , the computational time required (for the heuristic to terminate) equals 59% and 55% of the full optimal solution time, respectively.

Figure 5.6 presents graphically the results of Table 5.4 with respect to the computational time needed and the quality of the solution obtained for different values of  $\lambda$  using the  $P + 1$  method. In this Figure: (a) the computational time is normalized with respect to the computational time of the full B&P method (100%), and (b) the cost difference is the average

percent cost difference with respect to the optimal integer solution. (Again for the cost difference we considered only the instances for which the optimal integer solution was not obtained).

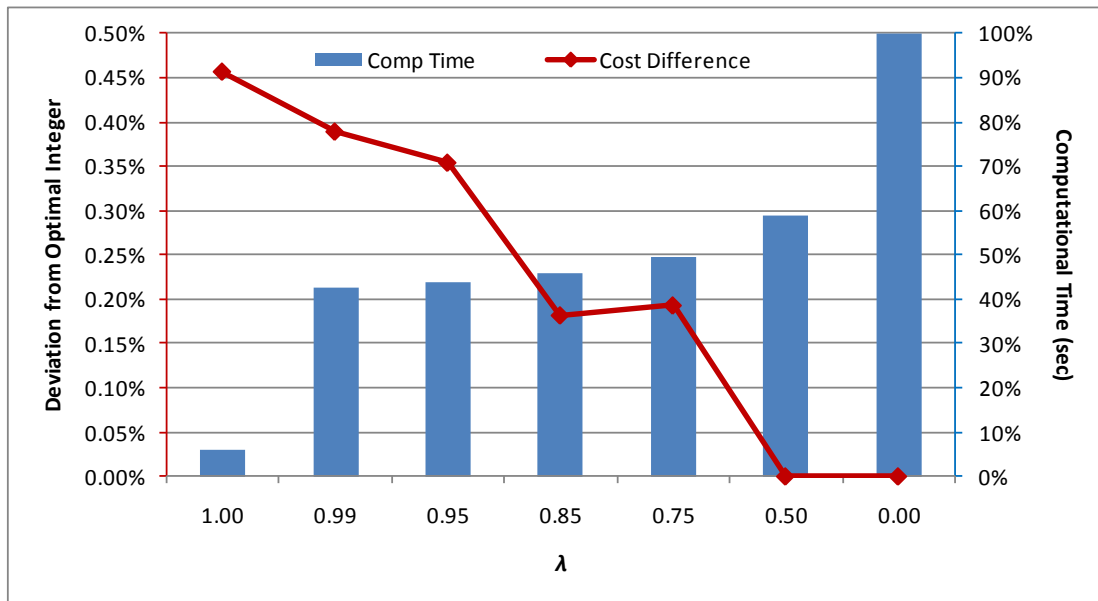


Figure 5.6: Computational times and deviation from optimal integer solution for different values of  $\lambda$  ( $P + 1$  method)

The results obtained validate the efficiency of the heuristic pruning technique, since

- The optimal integer solution is almost always obtained for  $\lambda = 0.5$  in approximately 60% of the time required by the full B&P
- The deviation of the solutions of the heuristic from the optimal integer solutions are very limited and controllable by the value of  $\lambda$ . Even for  $\lambda = 1$ , the cost deviation is less than 0.5%, while the time savings are 94% with respect to the full B&P.



## Chapter 6:      ENHANCEMENTS FOR APPLYING THE MPVRPTW IN A ROLLING HORIZON FRAMEWORK

In this Chapter we discuss important enhancements that are required in order to address problems of practical significance using the MPVRPTW methods presented in Chapters 4 and 5. Specifically, we focus on the following issues:

- Employing MPVRP in rolling horizon planning (see Section 6.1).
- Developing fundamental understanding of rolling horizon planning using a special quasi-static case (see Section 6.2)
- Addressing the case in which not all customer orders can be satisfied within the planning horizon (see Section 6.3).
- Testing the MVRP in a rolling horizon environment for both the special quasi-static case, and the more general dynamic case (see Section 6.4).

### 6.1 THE ROLLING HORIZON PLANNING PROCESS

In order to be able to address long-term horizons using the MPVRPTW we utilize a rolling horizon framework in which the solution procedure of the MPVRPTW is embedded (see Fig. 6.1). Consider an environment in which each customer order  $i$  arriving in period  $t$  can be served within a period window  $[\xi_i^s, \xi_i^e]$ , where  $\xi_i^s > t$ ; that is, the order may be served after (and not including) the period of arrival. The *long term time horizon* defined by the latest expiration time of any unserved order,  $\xi_i^e, \forall i \in N$  (where  $N$  is the set of unserved orders) is denoted as  $S$ . Note that  $S$  depends on  $N$ , but for simplicity this is not indicated explicitly in the  $S$  symbol.

Denote the current period as  $p_c$  and assume that we have elected to solve MPVRPTW for the period interval  $[p_c + 1, p_c + P]$ , where  $p_c + P \leq S$ . This latter time interval is the *planning horizon*. The fact that we consider a planning horizon  $[p_c + 1, p_c + P]$ , which is a subset of the long term horizon  $S$  may be attributed to the following reasons:

- The available information on customer orders to be serviced in distant periods from the current period may be limited, and, the period windows of these customer orders may not be overlapping with the customer orders within the selected planning horizon.

Thus, there is no point in considering these periods within the planning horizon, since they will not affect the planning of the selected planning horizon.

- The horizon  $S$  may be long, so that solving the MPVRPTW over  $S$  may be computationally intractable, due to the associated large number of orders.

Let  $MPVRPTW(P, p_c + 1)$  denote the multi period routing problem to be considered over the planning horizon  $[p_c + 1, p_c + P]$  of length  $P$ . The customer orders to be considered in this problem (set  $\bar{N}$ ) are those for which the period window has opened within the planning horizon; i.e.  $\bar{N} = \{i \in N: p_c + 1 \leq \xi_i^S \leq p_c + P\}$ , and  $(\bar{N} \subseteq N)$ .

Given this set up, the related planning process is as follows: Customer orders are assigned over the next  $P$  periods  $[p_c + 1, p_c + P]$  using  $MPVRPTW(P, p_c + 1)$ . The orders assigned in periods  $[p_c + 1, \dots, p_c + M]$ , where  $M \leq P$ , are selected for service. The remaining orders, assigned in the time interval  $[p_c + M + 1, p_c + M + P]$  are considered again for routing combined with the new customer orders that arrive during the execution of periods  $p_c + 1, \dots, p_c + M$ . This rolling (planning) horizon process is shown in Fig. 6.1, where  $M$  is the length of the implementation period.

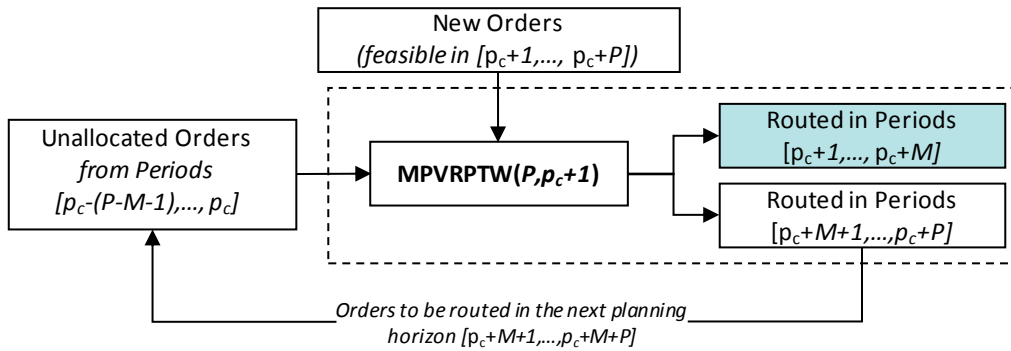


Figure 6.1. Planning process

The length  $P$  of the planning horizon is selected in order to balance the quality of the combinations, and resulting routes, formed by the known customer orders within the horizon, versus the computational effort required to solve the problem.

## 6.2 THEORETICAL INSIGHTS FOR A SPECIAL CASE OF ROLLING HORIZON PLANNING

### 6.2.1 NOTATION

Consider a routing problem over a long term horizon of  $S$  periods, in which all customer orders are known throughout the horizon. Let also the sole objective of this problem be the minimization of the routing cost.

Consider, now, solving the above problem by a rolling horizon scheme with planning horizon of length  $P$  and implementation horizon of length  $M$ . Thus, the rolling horizon cycle will be repeated every  $M$  periods. We call this case “quasi-static”, since each time we solve a MPVRPTW, the only new clients considered are those of the last  $M$  periods of the planning horizon. No new customer orders arrive dynamically. Using this special case, we will develop some interesting theoretical insights. Prior to this, we will introduce necessary notation.

Let  $MP(P, p_c + 1)$  be the optimal solution of the multi-period problem  $MPVRPTW(P, p_c + 1)$ , and  $\mathcal{C}(P, p_c + 1)$  be the related optimal cost; that is the cumulative routing cost considering all periods of the planning horizon:  $\mathcal{C}(P, p_c + 1) = \sum_{\omega=p_c+1}^{\min(p_c+P, S)} C(P, \omega)$ , where  $C(P, \omega)$  denotes the routing cost of period  $\omega$ . Given this notation, we denote as  $\bar{C}_{PM}^S$  the final cost of the entire long-term horizon plan for a planning horizon of  $P$  periods and an implementation horizon of  $M$  periods. For convenience we assume that  $P$  is an integer multiple of  $M$ . Then,

$$\bar{C}_{PM}^S = \sum_{p=1}^{\frac{S}{M}} \sum_{k=1}^M C(P, p * M + k) \quad (6.1)$$

Consider any set of feasible customers within period  $p$ , and the optimal w.r.t. cost solution for serving these customers. If there are multiple optimal solutions for this set of customers, we arbitrarily select one. We denote as  $O_p$  the set that contains the optimal solutions (i.e. combinations of routes) for all subsets of customers that are feasible within period  $p$ . Note that for each different customer subset there is only one solution in  $O_p$ .

### 6.2.2 THEORETICAL INSIGHTS

In the current Section we propose important statements regarding rolling horizon planning for long term quasi-static routing problems.

The first statement compares the monolithic solution of the full routing problem (for the  $S$  period horizon) with any solution obtained by a rolling horizon scheme.

### Statement 1

*Given that all customer orders to be served within the long-term horizon ( $S$  periods) are known, the cost  $\bar{C}_{SM}^S$  of the optimal solution of  $MPVRPTW(S, p_c + 1)$  is always lower than or equal to the final implemented cost  $\bar{C}_{PM}^S$  obtained by any rolling horizon scheme with planning horizon of  $P < S$  periods, and implementation horizon  $M$ .*

The justification of this statement has as follows:

Consider the solution of a problem provided by a rolling horizon scheme, which solves a sequence of problems  $MPVRPTW(P, p_c + 1)$  for  $p_c = 0, M, 2M, \dots, S - M$ . The current period is considered to be  $p_c = 0$ . This solution, with cost  $\bar{C}_{PM}^S$ , is also a feasible solution of the monolithic problem  $MPVRPTW(S, 1)$ . The justification of this has as follows:

- The feasible space of  $MPVRPTW(S, 1)$  may be formed by considering (a) all feasible distributions of customer orders among the periods of horizon  $S$ , and (b) for each period all feasible routes of the corresponding set of customer orders.
- Consider the optimal solution of  $MPVRPTW(S, 1)$ . For each period  $p$  of horizon  $S$ , the optimal solution contains the optimal routes of the customer orders allocated to that period; these routes belong to set  $O_p$  defined in the previous Section. If we denote by  $\check{O}(S, \omega)$  the set of routes of period  $\omega$  within the solution of  $MPVRPTW(S, 1)$ , then  $\check{O}(S, 1) \in O_1, \dots, \check{O}(S, S) \in O_S$ .
- Consider now, the solution of the same problem derived by a rolling horizon scheme with planning horizon of  $P$  periods, and denote by  $\check{O}(P, \omega)$  the set of routes of period  $\omega$  that belong to this solution. Since the solution of each  $MPVRPTW(P, p + 1)$  is affected (constrained) by the part of the solution implemented up to period  $p$ , it holds that the solution of the first period  $\check{O}(P, 1) \in O_1$ , while the solutions of subsequent periods belong to subsets ( $O'_p$ ) of each  $O_p$ , e.g.  $\check{O}(P, 2) \in O'_2 \subseteq O_2, \dots, \check{O}(P, S) \in O'_S \subseteq O_S$ .

Based on the above, the resulting cost  $C(S, \omega)$  of each period  $\omega$ , which corresponds to the set of routes  $\check{O}(S, \omega)$ , is always less than or equal to the cost  $C(P, \omega)$  of  $\check{O}(P, \omega)$ , and, ; therefore, the cost  $C_{SM}^S = \sum_{p=1}^{S/M} \sum_{k=1}^M C(S, p * M + k)$  of the optimal solution  $MP(S, 1)$  is always lower

than or equal to the cost  $C_{PM}^S$  obtained by the rolling horizon solution with planning horizon of  $P$  periods ( $C_{SS}^S \leq \bar{C}_{PM}^S$ ).

### Example 1

To illustrate this fact, consider a simple example with 2 periods only, and two different planning horizons ( $P = 1$  and  $P = 2$ ). Let also the implementation horizon be equal to one ( $M = 1$ ).

- For the case with  $P = 1$ , two single period problems are solved, and the final cost of the two-period problem is  $\bar{C}_{11}^2 = C(1,1) + C(1,2)$
- For the case with  $P = 2$ , one two-period problem is solved, and the final cost is given by  $\bar{C}_{21}^2 = C(2,1) + C(2,2)$ .

For period 1, the feasible routes are the same for both cases. For period 2, however, and for the case with  $P = 1$ , the feasible set of routes, along with the routing cost  $C(1,2)$  is restricted by the routing result of the first period problem  $MP(1,1)$ . Thus, for  $P = 2$  the feasible set of period 2 is the set  $O_2$  of optimal solutions of all sets of customer orders that are feasible in period 2 (see Section 6.2.1 above). For  $P = 1$  the feasible set of period 2 is a subset of  $O_2$ , since the available set of feasible solutions has been restricted by the solution of period 1. Thus, every solution obtained by the planning horizon  $P = 1$  can also be obtained by  $P = 2$ , and the cumulative cost  $C(2,1) + C(2,2)$  is less than or equal to  $C(1,1) + C(1,2)$ .

Statement 2 below is related to the length  $P$  of the planning horizon of a rolling horizon scheme that solves the long term problem of  $S$  periods (quasi-static case). One may assume that a longer planning horizon may provide more efficient solutions, since it allows for an increased number of customer combinations, thus leading to the formation of more efficient routes. Statement 2 indicates that this is not necessarily true. The quality of the solutions is strongly related to the period flexibility of the customers and their characteristics (time windows, etc).

### **Statement 2**

*Consider the quasi-static routing problem of  $S$  periods. The overall routing cost ( $\bar{C}_{PM}^S$ ) provided by a rolling horizon scheme with planning horizon of  $P$  periods is not necessarily lower than or equal to the overall routing cost ( $\bar{C}_{P'M}^S$ ) provided by a rolling horizon scheme with planning horizon of  $P'$  periods, where  $P' < P < S$  for the same  $M$ .*

This fact is illustrated by the following example (and verified experimentally in Section 6.4).

### Example 2

Consider five consecutive periods and four customers with period windows as shown in Table 6.1. Additionally, we consider that in each period only two customers may be routed and  $M$  is considered equal to 1. The related distances are provided in Table 6.2.

Table 6.1: Customers and related period windows (example 2)

Customer	Period Window
$a$	[1,2]
$b$	[2,4]
$c$	[4,5]
$d$	[5,5]

Table 6.2: Costs of arcs (example 2)

Arc	Cost	Arc	Cost
$(a, b)$	0.5	$(D, a)$	1.0
$(b, c)$	0.4	$(D, b)$	1.0
$(c, d)$	0.5	$(D, c)$	1.0
		$(D, d)$	1.0

We compare two alternative planning horizons,  $P = 2$  and  $P = 3$ . Fig. 6.2 and Table 6.3 present the final routes that will be generated from the alternative planning horizons. The relevant arc costs are shown in Fig. 4.

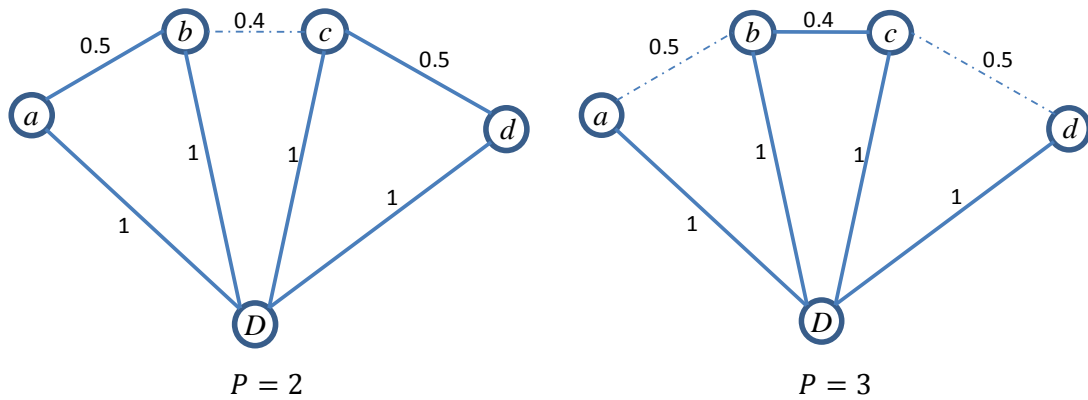


Figure 6.2: Customers and related network (example 2)

Table 6.3: Final routes per period for implementation horizon of  $P = 2$  and  $3$  (example 2)

Period	$P = 2$		$P = 3$	
	Routes	Cost	Routes	Cost
1	-	-	-	-
2	[D-a-b-D]	$(D, a) + (a, b) + (b, D)$	[D-a-D]	$2 \times (D, a)$
3	-	-	-	-
4	-	-	[D-b-c-D]	$(D, b) + (b, c) + (c, D)$
5	[D-c-d-D]	$(D, c) + (c, d) + (d, D)$	[D-d-D]	$2 \times (D, d)$
<b>Total</b>		<b>5</b>		<b>6.4</b>

The final routing costs of each planning horizon are:

- $P = 1$ :  $\bar{C}_{2,1}^5 = (D, a) + (a, b) + (b, D) + (D, c) + (c, d) + (d, D) = 5.0$
- $P = 2$ :  $\bar{C}_{3,1}^5 = 2(D, a) + (D, b) + (b, c) + (c, D) + 2(D, d) = 6.4$

Thus, for this example, the planning horizon of 2 periods results in better (lower) routing cost. Note, although, that with appropriate period windows, this situation may be reversed.

Statement 3 concerns the length of the implementation horizon  $M$  of a rolling horizon scheme for the quasi-static case. In practice it is typical to use the minimum possible  $M$  (i.e.  $M = 1$  in our case). Note that if  $M > 1$  the step of the rolling horizon scheme is modified appropriately to match  $M$ .

Implementing only the first period of the solution  $MP(P, p_c)$  may seem the most appropriate tactic, due to the fact that there is no knowledge of the customer orders beyond period  $p_c + P$ . Thus, implementing the minimum possible part of the solution may offer the opportunity to incorporate in a better fashion the new orders of the next problem to be solved. This, however, turns out not to be necessarily true.

### Statement 3

*Consider the quasi-static case of the long term problem ( $S$  periods). If this problem is solved by a rolling horizon scheme with planning horizon  $P > 1$ , it is not guaranteed that  $M = 1$  (i.e. implementing only the part of the solution corresponding to the first period of the planning horizon) will lead to the minimum cost value  $C_{PM}^S$ .*

This fact is illustrated by the following example.

### Example 3

Consider the following problem with four periods and four customers with the period windows provided in Table 6.4. Additionally, we consider that in each period only two customers may be routed, and  $P$  is equal to 2. The related distances are provided in Table 6.5.

Table 6.4: Customers and related period windows (example 3)

Customer	Period Window
$a$	[1,2]
$b$	[2,3]
$c$	[3,4]
$d$	[4,4]

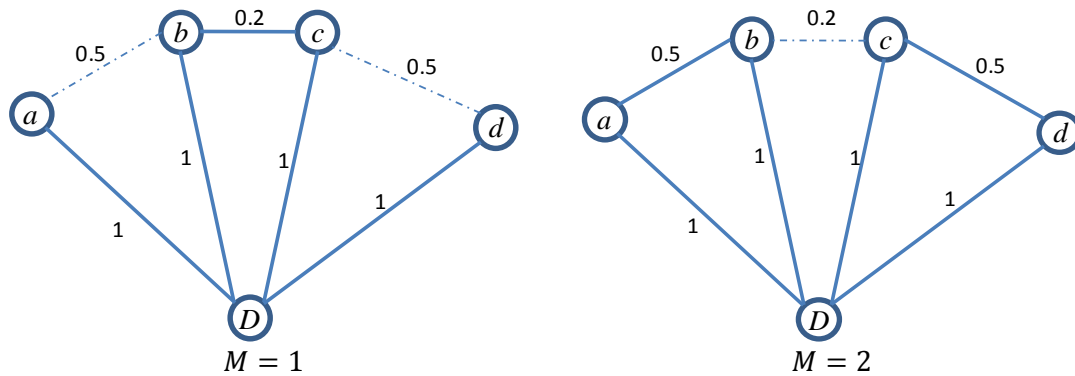
Table 6.5: Costs of arcs (example 3)

Arc	Cost	Arc	Cost
$(a, b)$	0.5	$(D, a)$	1.0
$(b, c)$	0.2	$(D, b)$	1.0
$(c, d)$	0.5	$(D, c)$	1.0
		$(D, d)$	1.0

Based on this information, Fig. 6.3 and Table 6.6 present the final routes that will be generated by using two different implementation horizons,  $M = 1$  and  $M = 2$ .

For  $M = 1$ : Since  $P = 2$ , in the first planning step of the first two periods only the first two customers are considered. Since customer  $b$  cannot be routed prior to period 2, both customers are planned for routing within the second period in order to minimize the routing cost. Thus, there are no clients routed in period 1. The second planning step considers periods 2 and 3 and customers  $a, b$  and  $c$ . Since distance  $(b, c)$  is lower than  $(a, b)$ , customers  $b$  and  $c$  are planned to be served together in period 3. Customer  $a$  is served alone in period 2 and is implemented during this second step. The next and final planning step considers periods 3 and 4 along with customers  $b, c$  and  $d$ . Similarly, in period 3 customers  $b$  and  $c$  are scheduled together in period 3 and customer  $d$  remains alone in period 4.

For  $M = 2$ : Using the same procedure but with implementing both planned periods, the final routes per period are shown in Table 6.6.

Figure 6.3: Customers and routes for implementation horizons  $M = 1$  and  $2$  (Example 3)Table 6.6: Routes and cost per period for implementation horizons  $M = 1$  and  $2$  (Example 3)

Period	$M = 1$		$M = 2$	
	Routes	Cost	Routes	Cost
1	-	-	-	-
2	[D-a-D]	$2 \times (D, a)$	[D-a-b-D]	$(D, a) + (a, b) + (b, D)$
3	[D-b-c-D]	$(D, b) + (b, c) + (c, D)$	[D-c-d-D]	$(D, c) + (c, d) + (d, D)$
4	[D-d-D]	$2 \times (D, d)$	-	-
<b>Total</b>		<b>6.2</b>		<b>5.0</b>

The final routing cost of each implementation horizon are:

- $M = 1: \bar{C}_{1,1}^4 = 2(D, a) + (D, b) + (b, c) + (c, D) + 2(D, d) = 6.2$
- $M = 2: \bar{C}_{1,2}^4 = (D, a) + (a, b) + (b, D) + (D, c) + (c, d) + (d, D) = 5.0$

For this example using an implementation horizon with  $M = 2$  results in lower routing costs if  $(D, a) + (b, c) + (D, d) > (a, b) + (c, d)$ ; this in our example holds for appropriately large values of the arcs connecting the depot and the customers.

### 6.3 MODIFYING MPVRPTW TO DEAL WITH LIMITED RESOURCES AND UNSERVED CUSTOMERS

When addressing multi-period routing problems with a limited number of vehicles, not all customer orders can be routed within the selected planning horizon. Thus, the following important issues need to be considered:

- How to deal with cases in which not all customers can be served with planning horizon  $P$ , due to resource limitations?
- Which customers to exclude from the current plan in that case?
- Even if the resources are adequate, depending on the problem model (e.g. the objective function), customers may be excluded in order to save routing costs; how can one deal with this matter?

An additional important issue to be considered when using a rolling horizon framework within a multi-period setting, is the following: How to deal with the tendency of the rolling horizon to postpone the scheduling of customers; this results in customers the period window of which expires in the first period of the planning horizon (e.g. customers with  $\xi_i^s = p_c + 1$ ).

These issues are relevant when a rolling horizon framework is used to solve long term routing problems (both quasi-static and dynamic). In this case, customers excluded from the solution of the problem solved for a certain planning horizon will be considered by the problem(s) corresponding to subsequent planning horizon(s) (if allowed by the customers' period windows). This fact tends to “push” customers into the future, and may lead to unserved customers, due to resource or period window constraints.

This situation describes cases, in which certain customers are selected against other customers, and the latter remain unserved due to resource limitations. In the subsequent planning horizons the period windows of these unplanned customers are becoming

progressively narrower, leading to more “expiring” customers. In case not all customers can be accommodated, the selection of which (expiring) customers to be left unserved is based strictly on routing costs.

Below, the methods described in Chapters 4 and 5 for the MPVRPTW are further enhanced in order to address the aforementioned issues.

### **6.3.1 ENHANCING THE OBJECTIVE FUNCTION**

As indicated previously, the objective function of MPVRPTW accounts strictly for the routing cost (see Chapters 3 and 4) and it regards cases in which all customers can be routed within the selected planning horizon (i.e. with enough resource capacity to facilitate all orders). For the case addressed in this Section, customers may be left unserved due to limited resources (time, demand, and limited fleet). Two related issues arise then: (a) Since the model proposed in Chapter 3 includes the constraint to serve all customers, and, if no adequate resources are available, then there is no feasible solution; (b) on the other hand, if the constraint for serving all customers is dropped, then customers may not be included in the solution solely based on routing cost (e.g. remote customers). To address these related issues we have introduced additional (penalty) terms in the objective function to prevent dropping selected clients. We have also used a way to artificially satisfy the constraint of serving all customers.

Consider the case of using MPVRPTW in a rolling horizon setting. If the period window of an unserved customer is such that the latter may be re-planned in the next planning horizon, then dropping the said customer may not become an issue, since this customer will be considered again in the subsequent problem(s). If, on the other hand, the period window of the customer expires in the first period of the current planning horizon, then the customer will be left unserved, resulting to a severely negative impact to customer service. The proposed penalty function should take this fact into account and avoid dropping such “expiring” customers.

Note that for simplicity, and without loss of generality, below and in the following Sections we consider that  $p_c = 0$ ; in this case the above “expiring” customers should be served within the first period of the planning horizon.

Consider the solution of a MPVRPTW in the planning horizon  $[1, P]$  and let  $u_e$  and  $u_f$  be the sets of expiring and non-expiring customers (derived from the feasible set of customers  $N$

within the planning horizon), which are left unassigned due to resource limitations. We propose the following straightforward modification of the objective function in order to simultaneously:

- (a) Maximize the number of customers served in all periods of the planning horizon,
- (b) Maximize the number of expiring customers served in period 1,
- (c) Minimize the multi-period routing cost.

$$\min \left( \sum_{p=1}^P \sum_{r \in \Omega_p} C_r^p x_r^p \right) + P_e |u_e| + P_f |u_f| \quad (6.2)$$

where  $P_e$  and  $P_f$  are the penalties for each expiring or non-expiring unassigned customer, respectively. In order to minimize the number of unserved customers, both penalties need to be set to large values. These penalties can be set to any value larger than  $\max_{i \in N} (C_{r_i})$  where  $C_{r_i}$  represents the cost of the *unit* route  $[Depot - i - Depot]$ . If the penalties are lower than the unit route costs ( $C_{r_i}$ ), there is possibility that in column generation the artificial routes – columns related to the penalties (e.g. leaving a customer unserved) may enter the final basis of the solution as opposed to the actual unit routes (however the grouping of customers to routes might prevent this behavior).

Note that the proposed modification of Eq. (6.2) needs to be made only to the objective function of the Master Problem, Eq. (3.14). Thus,

- Each unserved expiring or non-expiring customer is allocated to a *virtual* unit route  $[Depot - i - Depot]$  with artificial routing cost equal to  $P_e$  or  $P_f$ , respectively. Thus, Constraints (3.16) (e.g. each customer should be served once) still remain feasible by this artificial assignment.
- The elements of these artificial columns (routes) that are relevant to the vehicle Constraints (3.15) are all equal to zero in order to not contribute to the number of used vehicles. Note that in a route (column) assigned to a specific period the relevant element of the vehicle constraint is equal to one in order to consider that one vehicle is used by the solution.
- Although, the unserved customers contribute to the total final cost, by the assigned penalties, they do not interfere with the actual routing costs, since the routing costs are provided by the sequence of the assigned customer orders to each proposed route (column) within the Master Problem.

### 6.3.2 ADJUSTING THE PENALTIES $P_e$ AND $P_f$ TO PRIORITIZE EXPIRING CUSTOMERS

In order to ensure that an expiring customer will not be replaced by a non-expiring customer, we consider using the following inequality

$$P_e \geq P_f + \Delta M_{e,f} \quad (6.3)$$

and we identify the appropriate values of  $\Delta M_{e,f}$  to ensure that expiring customers will be served, within the limits imposed by the resource constraints. That is, expiring customers may still remain unserved in cases in which the available resources are not adequate to facilitate them.

In order to compute the values for  $\Delta M_{e,f}$ , we consider a theoretical multi-period routing problem with two different solutions (1) and (2), the routing costs of which are given by  $R^{(1)}$  and  $R^{(2)}$ , respectively, and  $|u_e^{(1)}| < |u_e^{(2)}|$ . In order for the objective function (6.2) to provide lower cost for solution (1) in comparison to solution (2), the following should hold:

$$R^{(1)} + |u_e^{(1)}| P_e + |u_f^{(1)}| P_f < R^{(2)} + |u_e^{(2)}| P_e + |u_f^{(2)}| P_f \quad (6.4)$$

If we substitute  $P_e$  with  $P_e = P_f + \Delta M_{e,f}$ , then (6.3) becomes:

$$\Delta M_{e,f} > \frac{(R^{(1)} - R^{(2)}) + (u_f^{(1)} - u_f^{(2)}) P_f}{(u_e^{(2)} - u_e^{(1)})} - P_f \xrightarrow[\Delta u_e^{(2,1)} = u_e^{(2)} - u_e^{(1)}]{\substack{\Delta R^{(1,2)} = R^{(1)} - R^{(2)} \\ \Delta u_f^{(1,2)} = u_f^{(1)} - u_f^{(2)}}} \quad (6.5)$$

$$\Delta M_{e,f} > \frac{\Delta R^{(1,2)} + \Delta u_f^{(1,2)} P_f}{\Delta u_e^{(2,1)}} - P_f \quad (6.6)$$

Thus, in order for (6.3) to hold, we should define  $\Delta M_{e,f}$  to always be greater than the right hand side of the inequality.

- The worst case of  $\Delta R^{(1,2)}$  is when  $R^{(1)} = \sum_{i=1}^N C_{r_i}$ , and when  $R^{(2)} = 0$  (i.e. none of the customers are visited). Thus  $\Delta R^{(1,2)} \leq \sum_{i=1}^N C_{r_i}$
- The worst case for  $\Delta u_f^{(1,2)}$  is when  $|u_f^{(1)}| = \hat{f}$ , where  $\hat{f}$  is the number of non-expiring customers, and  $|u_f^{(2)}| = 0$ . Thus,  $\Delta u_f^{(1,2)} \leq \hat{f}$ .
- Similarly,  $\Delta u_e^{(2,1)} \geq 1$ , since we have considered that  $|u_e^{(1)}| < |u_e^{(2)}|$  and  $|u_e|$  is always a positive integer.

Thus, by replacing the aforementioned equations to (6.5):

$$\frac{\Delta R^{(1,2)} + \Delta u_f^{(1,2)} P_f}{\Delta u_e^{(2,1)}} - P_f \leq \frac{\sum_{i=1}^N C_{r_i} + \hat{f} P_f}{1} - P_f = \sum_{i=1}^N C_{r_i} + (\hat{f} - 1) P_f \Rightarrow \quad (6.7)$$

$$\Delta M_{e,f} \geq \sum_{i=1}^N C_{r_i} + (\hat{f} - 1) P_f \quad (6.8)$$

Now since  $P_f > \max_{i \in N}(C_{r_i})$ , the difference (say  $\delta$ ) between the weights of the expiring and the flexible customers is given by the following inequality ( $\vartheta$  represents a small positive value):

$$\Delta M_{e,f} \geq \delta = \sum_{i=1}^N C_{r_i} + (\hat{f} - 1) \max_{i \in N}(C_{r_i}) + \vartheta \quad (6.9)$$

If  $\delta$  is set to this value, then the number of expiring customers that will be served is guaranteed to be maximized. Note that for short planning horizons, lower values of  $\Delta M_{e,f}$  result in a larger number of unserved customers.

### 6.3.3 ADJUSTING THE PENALTIES $P_e$ AND $P_f$ FOR ALL (EXPIRING AND NON-EXPIRING) CUSTOMERS

Using the aforementioned penalties we do not distinguish among *non-expiring* customers based the imminence of their expiration periods. Thus, the non-expiring customers to be assigned in period 1 will be selected solely based on their routing cost efficiency. This may lead to a myopic assignment of customers, without taking into consideration their flexibility (as defined by their expiration period), and may result in leaving customers with low flexibility unserved. The requirement of serving these customers in the next period(s) may increase the routing cost far beyond the savings incurred by excluding them from the original planning horizon.

To moderate this issue, we propose five (5) alternative penalty functions that provide the exclusion penalty of each customer depending on the imminence of its expiration period. That is, the penalty  $P_i^\gamma$  assigned to customer  $i$  depends on the customer's expiration period ( $\xi_i^e$ ) and on the shape of the penalty function  $\gamma$ . Note that this function provides the penalties for all customers, expiring and non-expiring, taking into account (for  $\gamma = 2, \dots, 5$ ) the analysis presented above for the difference of penalty values between the two types of customers. Each

of the penalty functions represents an alternative trade-off between routing costs and service level (maximization of served customers).

#### Penalty function $\gamma = 1$ : Ignoring the Period Window

This penalty function assigns the same penalty to all customers regardless their expiration period; thus,  $\Delta M_{e,f} = 0$  and  $P_f = \max_{i \in N} (C_{r_i}) + \vartheta$ . This has the following implications: (a) Expiring customers are not treated with priority in case resource limitations prevent all customers to be routed within their period windows and the planning horizon considered, and (b) routing costs are favored since the objective function allows the selection between customers to yield more efficient routes (i.e. the customers that result in the minimum routing costs).

$$P_i^1 = P_f \quad \forall i \in N \quad (6.10)$$

#### Penalty function $\gamma = 2$ : Forcing the Inclusion of Expiring Customers

This penalty function respects the limitations of expiring customers. That is, the large penalty value of Section 6.3.2 is assigned to penalty  $P_e$  for each expiring customer. All other customers, regardless their expiration period, are treated equally with the same penalty value  $P_f$ .

$$P_i^2 = \begin{cases} P_f + \delta & \forall i \in N: \xi_i^e = 1 \\ P_f & otherwise \end{cases} \quad (6.11)$$

As a result, in every multi-period problem, all expiring customers within the initial period of the planning horizon are routed, if this is possible. Expiring customers may still be left unrouted, but this is due only to resource limitations, and not to customer selection (i.e. in that case no non-expiring customer will displace an expiring one in the solution).

#### Penalty function ( $\gamma = 3, 4, 5$ ): Continuous Penalty Functions

In these functions, the penalty assigned to each customer  $i$  is based on each customer's expiration period ( $\xi_i^e$ ). Expiring customers in period 1 are assigned with a penalty of value  $P_e$ , while all others are assigned with a penalty value within the interval  $[P_f, P_e]$  according to the selected penalty function.

Note that in the case of penalty functions  $\gamma = 3, 4, 5$  expiring customers do not always displace non-expiring customers from the solution. This is because the penalties assigned to

the flexible customers increase as a function of the imminence of the customer's expiration date beyond the value that secures their displacement by the expiring customers (that is the difference between the penalties of expiring and non-expiring customers is not always greater than  $\delta$ ). Also,  $\max_{i \in N}(\xi_i^e) > 1$ . The three penalty functions are provided below:

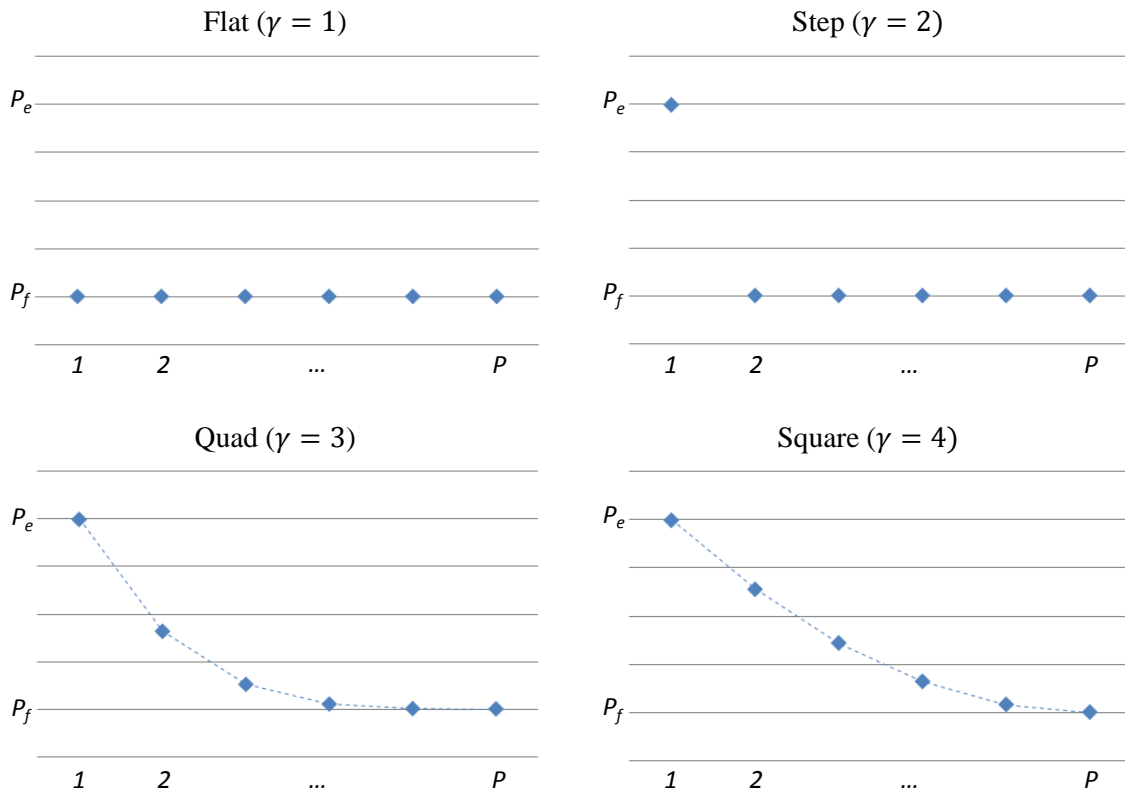
$$P_i^3 = \left( \frac{(\xi_i^e - 1)}{\max_{j \in N}(\xi_j^e) - 1} \right)^4 \times (P_f - P_e) + P_e \quad \forall i \in N \quad (6.12)$$

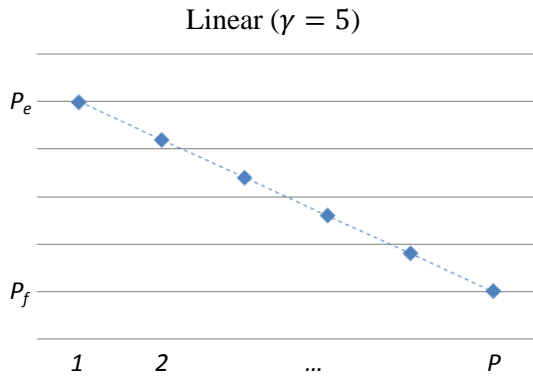
$$P_i^4 = \left( \frac{(\xi_i^e - 1)}{\max_{j \in N}(\xi_j^e) - 1} \right)^2 \times (P_f - P_e) + P_e \quad \forall i \in N \quad (6.13)$$

$$P_i^5 = \left( \frac{(\xi_i^e - 1)}{\max_{j \in N}(\xi_j^e) - 1} \right) \times (P_f - P_e) + P_e \quad \forall i \in N \quad (6.14)$$

Each function presents a quadratic, square and linear decrease of the penalty, respectively, with regard to each customer's expiration period ( $\xi_i^e$ ).

Figure 6.4 illustrates the five different penalty functions. By using the appropriate function, we may direct the solution method into prioritizing expiring customers, as well as customers with limited flexibility (i.e. available number of periods to be routed).



Figure 6.4: Penalty functions ( $\gamma = 1, \dots, 5$ )

### Insights on the above Penalty Functions

To gain some insight regarding the alternative penalty functions, a series of tests were conducted by solving multi-period routing problems with limited resources. The parameters taken under consideration in these tests are:

- The scheduling horizon was set to five (5) periods
- 2 vehicles were considered available per period of the planning horizon
- Only period window patterns 3 and 5 were considered, providing a moderate customer flexibility. Note that the following results consider both patterns cumulatively; more detailed results are provided in Appendix C.
- The planning horizon was set to  $P = 1$  to 5. For  $P < 5$ , a rolling horizon scheme was utilized in order to plan all five periods.
- When rolling horizon planning was used, only the first period of each planning horizon was implemented ( $M = 1$ ). The remaining customers (routed in periods 2 to  $P$ ) were considered again in the next planning horizon, until all periods were planned.

For the testing process we generated 10 test instances based on the R1 (R101, R102, R105, R109, R110) and C1 (C101, C105, C106, C107, C108) test sets of Solomon. For each test instance we selected the first 50 customers and distributed them in the scheduling horizon of the 5 periods (10 customers per period) in a sequential manner. Each instance was tested for each one of the proposed penalty functions ( $\gamma = 1$  to 5). For each planning horizon and each penalty function, the full B&P method was used providing optimal integer solutions for the related MPVRPTW. For some instances, namely, R102, R109, R110 and C108, and for period window pattern equal to 5, the B&P method did not reach the optimal solution within

the allowable time limit. Thus, the analysis included only 16 combinations of test instances and period window patterns for a total of 400 ( $16 \times 5 \times 5$ ) experiments.

In order to assess the performance of the different penalty functions, the following measures were defined:

- The percentage of unserved customers: For each  $P$  and  $\gamma$  combination we consider the total number of unserved customers w.r.t. the total number of customers over all 16 experiments corresponding to this combination
- The routing cost per served customer: The total implemented routing cost divided by the total number of served customers.

The results are shown in Table 6.7 and Figs. 6.5 to 6.7.

Table 6.7: Average test results for 16 test instances per penalty function and planning horizon

$P$	Unserved Customers					Routing Cost/Served Customer				
	$\gamma$					$\gamma$				
	1	2	3	4	5	1	2	3	4	5
1	12.6%	4.8%	4.4%	4.4%	4.9%	14.37	15.90	16.88	16.88	16.31
2	7.4%	4.4%	4.0%	4.0%	3.8%	13.42	14.86	15.28	15.40	14.90
3	5.4%	4.4%	3.9%	3.9%	3.6%	13.33	14.42	14.68	14.67	14.43
4	3.9%	4.1%	3.9%	3.9%	3.6%	13.62	14.43	14.56	14.56	14.32
5	3.8%	4.0%	3.9%	3.9%	3.6%	13.56	14.07	14.54	14.54	14.30
	<b>6.6%</b>	<b>4.3%</b>	<b>4.0%</b>	<b>4.0%</b>	<b>3.9%</b>	<b>13.66</b>	<b>14.74</b>	<b>15.19</b>	<b>15.21</b>	<b>14.85</b>

Figure 6.5 depicts the results of Table 6.7 w.r.t the ratio of unserved customers per planning horizon and penalty function. Figure 6.6 depicts the results w.r.t the unit routing cost per served customer.

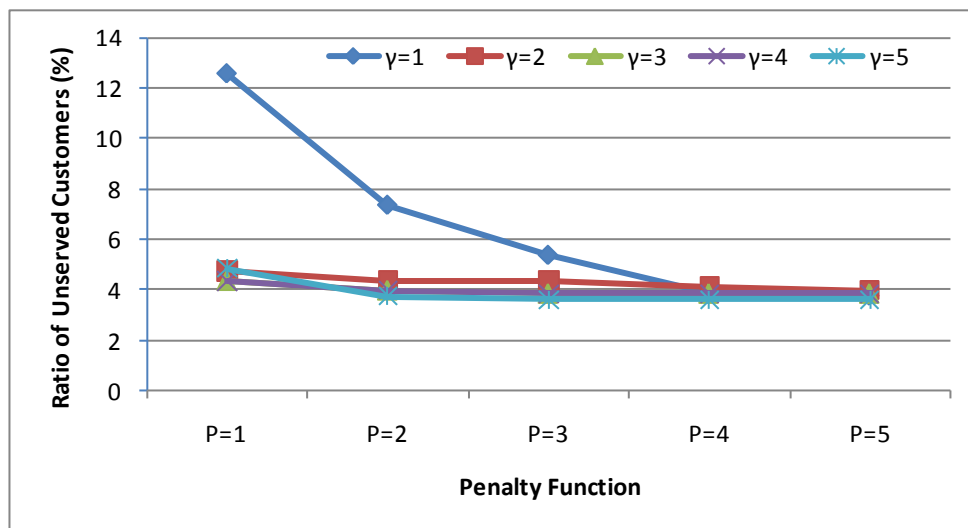


Figure 6.5: Ratio of unserved customers (%)

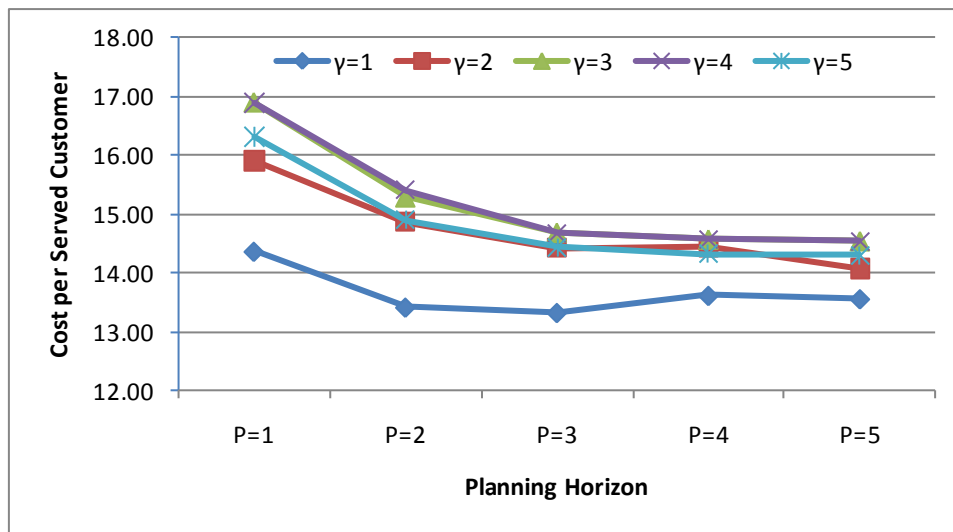


Figure 6.6: Cost per served customer

With respect to the length of the planning horizon, Table 6.7 and Fig. 6.5 illustrates that shorter planning horizons leave a larger number of customers unserved, for all penalty functions  $\gamma$ , though without the same effect among the various penalty functions. Note that as the planning horizons increase, there is more flexibility in assigning customers to periods.

With respect to the routing cost per served customer (Table 6.7 and Fig. 6.6), as expected, the larger planning horizons result in improved routing costs per served customer up to a certain value of  $P$  for all penalty functions.

Figure 6.7 presents the average ratio of unserved customers and the average cost per served customer across all planning horizons ( $P = 1$  to 5) for each penalty function ( $\gamma = 1$  to 5).

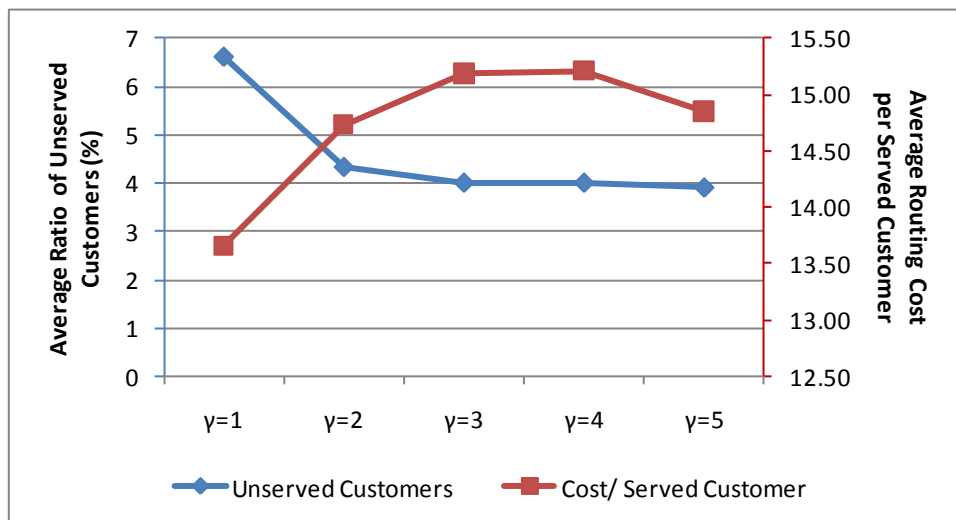


Figure 6.7: Unnerved customers and cost per served customer (all five penalty functions)

With respect to the penalty function, according to Fig. 6.7 penalty functions  $\gamma = 3, 4$  and 5 result in an increased number of served customers since they prioritize not only the expiring customers but also the soon-to-be expiring customers, thus, forcing the latter to be routed earlier, if possible. It is reasonable to expect that for these penalty functions the routing cost per served customer may increase, as shown in Fig. 6.7 for  $\gamma = 3, 4$ . Note that  $\gamma = 5$  results in the best routing costs, among these three penalty functions; it appears that the linear penalty function allows more flexibility in optimizing the routing costs. Overall, in a “Pareto optimal sense”, penalty function  $\gamma = 5$  seems to provide the most favorable results, and will be employed hereafter.

## 6.4 EXPERIMENTAL INVESTIGATION OF ROLLING HORIZON ROUTING

Having addressed issues of practical significance arising in rolling horizon planning, we have conducted an experimental investigation to study the significance of two critical parameters of such rolling horizon schemes: The length of the planning horizon  $P$ , and the length of the implementation horizon  $M$  (that is how many periods of the MPVRP. This study has focused on two distinct types of long term routing problems that differ in the degree of dynamism.

- The first problem type is the *quasi-static* problem introduced in Section 6.2. In this case, all customer orders, the period window of which starts within the planning horizon ( $p_c < \xi_i^s \leq p_c + P$ ), are considered to be known.
- In the second problem, customers become known at some point prior to the opening of their period window. In this case not all customer orders, the period window of which starts within the planning horizon, are known. Customer orders are revealed dynamically as time progresses. We call this problem, the *dynamic* rolling horizon routing problem.

The experimental testbed for both the quasi-static and the dynamic routing problems was constructed as follows:

- In terms of customer geographical distribution, we generated test instances of three types (random, clustered, and mixed), 3 test instances per type for a total of nine instances. The customer locations were selected from the extended Solomon benchmarks (Homburger and Gehring, 1999), which comprise 400 customers per problem
- Each test instance encompasses a long term horizon of 30 periods and 300 customers (i.e. the first 300 customers from the corresponding 400-customer extended Solomon problem)

- All customer coordinates were normalized in order to make the travel times (arc costs) of the extended Solomon benchmarks similar to the travel times (arc costs) of the Solomon benchmarks, based on the following scaling functions:

$$X = \frac{X_{gh} * X_{max}^{Sol}}{X_{max}^{gh}} \quad Y = \frac{Y_{gh} * Y_{max}^{Sol}}{Y_{max}^{gh}} \quad (6.15)$$

where  $X_{gh}$  is the actual x-coordinate from the Homberger and Gehring benchmarks,  $X_{max}^{gh}$  is the maximum x-coordinate from the same set and  $X_{max}^{Sol}$  is the maximum coordinate from the relevant problem type (i.e. R1, C1 and RC1) of the Solomon benchmarks (identical notation is used in the scaling function of the y-coordinate). This scaling allowed us to use the same time windows and service times for the customer data of the extended Solomon benchmarks, as those used in the original Solomon benchmarks

- For each instance, the characteristics of the customers (i.e., the time windows, the service times and the demands) were drawn from the relevant 100-customer Solomon benchmark, and were duplicated three times in order to obtain a total of 300 parameter sets. Each of these parameters, was allocated randomly and independently to the 300 customers of each corresponding instance
- Pattern 3 was used for the period window, since it provides moderate flexibility to customers in this multi period setting
- Two (2) vehicles were considered available per each period of the planning horizon in order to impose a strict limit on vehicle availability
- For each test instance, two implementation horizons ( $M = 1$  and  $M = 2$ ) and two planning horizons ( $N = 3$  and  $N = 5$ ) were tested initially
- The way customer orders are revealed varies per problem type:
  - In the quasi-static case, for each planning horizon we consider (only) those customers, of which their period window starts within the planning horizon. All these customers in the periods comprising the planning horizon are known. Thus, in each MPVRPTW the only new customer orders are those with period window starting at the last period of the planning horizon
  - In the dynamic case, we consider that each customer order becomes known one period prior to the opening of its period window. Thus, for each planning horizon starting at period  $p_c + 1$ , we consider the customer orders that have arrived (become known) up to period  $p_c$ .

The focus of our experimental investigation was to examine the effect of the planning horizon  $P$  and the implementation horizon  $M$  in rolling horizon route planning. This analysis focuses on two output attributes: (a) the customers served, and (b) the routing cost. In order to determine the effects of  $P$  and  $M$  on these attributes under a broad range of conditions, we used instances with a balanced mix of geographical distributions (R, C, RC) and different time window ranges (narrow, medium and wide). Thus, the results are considered to be unbiased with respect to these latter factors (geographical distribution and width of the time window).

The solution approach throughout the experimental investigation was based on the following:

- The rolling horizon scheme described above was utilized to solve each instance
- At each step of this scheme we solved the related MPVRPTW employing the following parameters:
  - The Cloning method was utilized for obtaining the lower bound
  - All column generation parameters remain the same as presented in Section 3.3.5
  - The linear penalty function was utilized (e.g.  $\gamma = 5$ ).
  - We implemented a Branch and Bound scheme on the columns generated by the CG algorithm while computing the lower bound of each MPVRPTW.

#### **Technical note: Transfer of routes within the first (implemented) period**

In cases in which the majority of customers have wide period windows, it is possible to identify routes in the final solution of a planning horizon, which are also feasible within the first period of the planning horizon (denoted as *Period1feasible* routes). If there are vehicles not used during this first period (have not been assigned to any route), then we select to transfer as many as possible of the said routes to the first period. That is, routes are assigned to period 1, starting from the *Period1feasible* route of the second period and moving to the subsequent periods. Transfer of routes terminates when all available vehicles of the initial period have been assigned to service.

### **6.4.1 EXPERIMENTAL RESULTS FOR THE QUASI-STATIC ROLLING HORIZON ROUTING PROBLEM**

Table 6.8 presents the results obtained for the above test instances in the quasi-static case. For each instance, the Table presents the number of customers routed and the average routing cost per customer, over all 30 periods for  $P = 3$  and 5, and  $M = 1$  and 2.

Table 6.8: Comparative results for different planning and implementation horizons (quasi-static case)

Instance	P	Routed Customers		Routing Cost per Customer	
		M=1	M=2	M=1	M=2
L_r103	3	293	287	20.09	20.62
	5	293	292	19.26	19.43
L_r106	3	299	298	19.54	20.03
	5*	299	289	18.31	18.17
L_r109	3	299	294	19.77	20.13
	5	299	293	17.62	18.27
L_c106	3	300	298	27.90	28.88
	5*	300	296	25.78	25.32
L_c108	3	300	300	22.72	24.41
	5*	300	297	21.68	21.59
L_c102	3	299	296	26.64	27.22
	5	299	298	24.40	24.52
L_rc101	3*	283	240	28.86	31.98
	5*	283	241	28.36	28.88
L_rc105	3*	293	253	26.84	27.95
	5*	294	267	25.52	27.28
L_rc107	3	300	298	23.35	23.76
	5	300	300	21.00	21.16
Average	3	296.2	284.9	23.97	25.00
	5	296.3	285.9	22.44	22.74
Time	Narrow	292.0	275.7	25.04	25.85
Windows	Medium	297.5	284.0	22.43	23.24
	Wide	299.3	296.5	22.13	22.51

\* Cases where implementation horizon  $M = 2$  results in lower routing cost compared to  $M = 1$ .

In terms of the implementation horizon  $M$ , it is clear that  $M = 1$  results in higher (or equal) number of routed customers in all cases. Furthermore, the value of  $M = 1$  results in lower routing cost per customer in almost all cases, except in the cases where  $M = 2$  resulted in a much lower number of served customers (also compare with Statement 3 of Section 6.2). These conclusions are validated by the average values of Table 6.8.

In terms of planning horizon  $P$ , the larger planning horizon ( $P = 5$ ) results in lower routing costs, having served slightly increased number of customers in comparison to  $P = 3$ . These conclusions are also validated by the average values of Table 6.8.

Since the data of Table 6.8 represent different problem types, with significant differences in geographical distribution and time window patterns, there are large variations in the final routing cost. In order to analyze the effect of planning parameters  $P$  and  $M$  over the different

problem types, we performed a paired difference t-test on the differences of the output parameters due to each of the selected factors (see Tables 6.9 and 6.10).

Table 6.9: Paired difference t-test for factor  $P$  (2 levels) -  $\alpha = 0,05$

	<i>Customers</i>			<i>Routing Cost</i>		
	<i>P=3</i>	<i>P=5</i>	<i>Dif.</i>	<i>P=3</i>	<i>P=5</i>	<i>Dif.</i>
<b>Mean</b>	290.56	291.11	-0.56	24.48	22.59	1.90
<b>Variance</b>	283.91	222.81	18.85	15.33	13.93	0.78
<b>Df</b>	17			17		
<b>t Stat</b>	<b>-0.54</b>			<b>9.13</b>		
<b>P(T&lt;=t) one-tail</b>	0.30			0.00		
<b>t Critical one-tail</b>	1.74			1.74		
<b>P(T&lt;=t) two-tail</b>	0.59			0.00		
<b>t Critical two-tail</b>	<b>2.11</b>			<b>2.11</b>		

The paired t-test of Table 6.9 for the effect of the planning horizon  $P$  indicates that:

- For the customers served, the mean difference ( $P = 3$  minus  $P = 5$ ) of the customers served is not significantly different than zero (t value,  $t(17) = -0.54$  and single tail probability 0.30). Thus, the increased planning horizon  $P = 5$  does not significantly increase the number of served customers, for a confidence level of 95%.
- On the other hand, for the cost per served customer, the mean difference ( $P = 3$  minus  $P = 5$ ) of is significant ( $t(17) = 9.13$  and single tail probability 0.00). Thus the observed improvement of the cost metric by increasing the planning horizon  $P = 5$  is significant.

Table 6.10: Paired difference t-test for factor  $M$  (2 levels),  $\alpha = 0,05$

	<i>Customers</i>			<i>Routing Cost</i>		
	<i>M=1</i>	<i>M=2</i>	<i>Dif.</i>	<i>M=1</i>	<i>M=2</i>	<i>Dif.</i>
<b>Mean</b>	296.28	285.39	10.89	23.20	23.87	-0.67
<b>Variance</b>	30.45	413.66	238.22	13.66	17.27	0.71
<b>Df</b>	17			17		
<b>t Stat</b>	<b>2.99</b>			<b>-3.36</b>		
<b>P(T&lt;=t) one-tail</b>	0.00			0.00		
<b>t Critical one-tail</b>	1.74			1.74		
<b>P(T&lt;=t) two-tail</b>	0.01			0.00		
<b>t Critical two-tail</b>	<b>2.11</b>			<b>2.11</b>		

The paired t-test of Table 6.10 for the effect of the implementation horizon  $M$  shows that:

- For the customers served, the mean difference ( $M = 1$  minus  $M = 2$ ) of the customers served is significantly different than zero ( $t(17) = 2.99$  and single tail probability 0.00).

Thus, implementation horizon  $M = 1$  increases significantly the number of served customers, for a confidence level of 95%.

- On the other hand, for the cost per served customer, the mean difference ( $M = 1$  minus  $M = 2$ ) is not significant ( $t(17) = -3.36$  and single tail probability 0.00). Thus, there is no significant difference among the two different implementation horizons concerning the routing cost, for a confidence level of 95%.

Figure 6.8 illustrates the variation of the cost per routed customer over the periods of the long term scheduling horizon for instance L\_R109. The Figure presents the results per each combination of planning and implementation horizon. Similar results have been obtained for all other instances. In this Figure, the cost ratio value in a certain period is the ratio of the total routing cost from period 1 till the period under consideration, divided by the total number of customers routed till the said period.

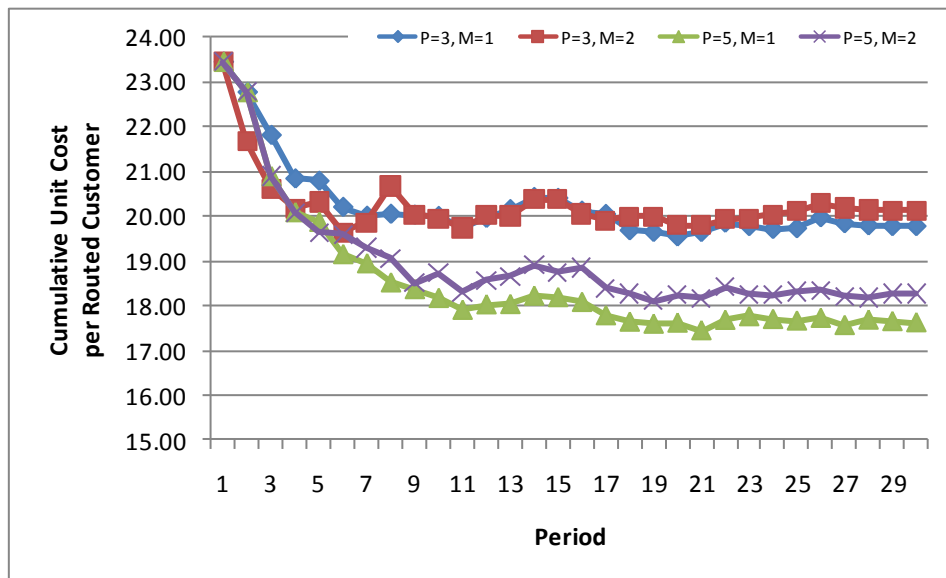


Figure 6.8: Cumulative unit cost per routed customer and period (L\_R109 Instance)

In Figure 6.8 the combination of  $M = 1$  and  $P = 5$  results in the best overall cost values throughout the long term horizon. The combination  $P = 5, M = 2$ , provides better results than the two remaining combinations; however due to the implementation horizon value ( $M = 2$ ) the number of served customers is reduced. Similar results are observed in the rest of the test instances as reported in Appendix D.1.

## 6.4.2 EXPERIMENTAL RESULTS FOR THE DYNAMIC ROLLING HORIZON ROUTING PROBLEM

As mentioned previously, in the dynamic rolling horizon routing problem each customer order becomes known a certain number of periods prior to the opening of its period window. In the experimental investigation of this problem we assume the (most stringent) case in which the order becomes known one period prior to the opening of the order's period window. Furthermore in the initial experiments:

- as before, two planning horizon values are used (3 and 5 periods), and,
- only the value  $M=1$  is used for the implementation horizon.

Note that  $M>1$  is not deemed appropriate in the dynamic problem; given that new customers arrive in each period, a value of  $M>1$  delays the planning of the arriving customers for  $M-1$  periods. This is illustrated in Table 6.11 which presents the related analysis for instance L\_r103 and a long term horizon of 30 periods. It is clear that  $M = 2$  results in significant fewer number of routed customers w.r.t  $M=1$ .

Table 6.11: Example illustrating the effect of the implementation horizon in the dynamic rolling horizon case

Instance	P	Routed Customers		Unit Cost per Customer	
		M=1	M=2	M=1	M=2
L_r103	3	295	164	18.80	20.77
	5	294	164	19.14	20.09

Table 6.12 presents the results of the experimental investigation, which was conducted based on the above parameters. The Table reports the instance name, the planning horizon used ( $P$ ), the total number of customers routed and the average routing cost per customer over the 30-period horizon. Appendix D.2 presents the detailed figures per instance and period of the planning horizon.

Table 6.12: Comparative results using different planning horizons (dynamic arrival of customers)

Instance	Planning Horizon			
	$P = 3$		$P = 5$	
	Routed Customers	Unit Cost/ Served Customer	Routed Customers	Unit Cost/ Served Customer
L_r103	295	18.80	294	19.14
L_r106	299	17.60	299	18.19
L_r109	299	17.91	299	17.91
L_c106	300	25.09	300	25.14
L_c108	300	24.68	300	24.98

<b>L_c102</b>	299	24.44	299	23.25
<b>L_rc101</b>	290	25.71	283	26.39
<b>L_rc105</b>	296	24.44	294	23.99
<b>L_rc107</b>	300	22.02	300	20.60

In terms of the planning horizon  $P$ , there is no strong evidence in favor of either value of  $P$ , neither in terms of the number of customers served nor in terms of routing cost per customer served. This is also shown by the results of the paired difference t-test presented in Table 6.13.

Table 6.13: Paired difference t-test for factor  $P$  (2 levels)

	<i>Customers</i>			<i>Routing Cost</i>		
	<i>P=3</i>	<i>P=5</i>	<i>Dif.</i>	<i>P=3</i>	<i>P=5</i>	<i>Dif.</i>
<b>Mean</b>	297.56	296.44	1.11	22.30	22.18	0.12
<b>Variance</b>	11.28	31.28	5.36	11.00	10.57	0.57
<b>Df</b>	8			8		
<b>t Stat</b>	<b>1.44</b>			<b>0.49</b>		
<b>P(T&lt;=t) one-tail</b>	0.09			0.32		
<b>t Critical one-tail</b>	1.86			1.86		
<b>P(T&lt;=t) two-tail</b>	0.19			0.64		
<b>t Critical two-tail</b>	<b>2.31</b>			<b>2.31</b>		

Based on the above results, it seems that wider planning horizons do not always succeed in improving routing costs. This may be attributed to the fact that wide planning horizons tend to spread customers over an increased number of periods without full knowledge of the future customers to appear.

#### **Dynamic rolling horizon case: A more detailed investigation of the effect of $P$**

In order to further examine the effect of the planning horizon, we conducted a series of additional tests with  $P$  varying from 1 to 7. In these tests we used a wider period window pattern (7 periods). In each test we selected the first 360 from the 400 customers of each extended Solomon instance, and used an arrival rate of 12 customers per period with each customer order becoming known one period prior to the opening of its period window.

Table 6.13 presents the average results per problem type. That is, the number of the served customers per period and the cost per customer has been averaged over the three test instances of the same problem type (random, clustered and mixed). Appendix D.3 presents the detailed results per instance.

Table 6.14: Average results per planning horizon and problem type (dynamic arrival of customers)

<i>P</i>	Problem Type					
	R1		C1		RC1	
	Served Customers	Unit Cost per Customer	Served Customers	Unit Cost per Customer	Served Customers	Unit Cost per Customer
1	356.7	21.2	360.0	34.8	347.0	24.2
2	358.7	19.1	360.0	25.6	349.7	22.8
3	357.3	16.9	359.7	22.5	347.0	21.5
4	305.3 <sup>(1)</sup>	15.5	358.7	20.8	348.7	20.6
5	356.0	15.3	359.3	20.8	347.3	20.0
6	356.0	15.8	358.3	20.5	348.7	19.4
7	342.0	15.9	359.3	21.7	344.0	20.1

(1) Low number of served customers is due to premature termination of the solution procedure at period 17.

Figure 6.9 displays the average unit cost per routed customer for all planning horizon values, and for the three customer distribution types. Note that the served customers present a slight but not considerable decrease as the planning horizon increases (see Figure 6.10).

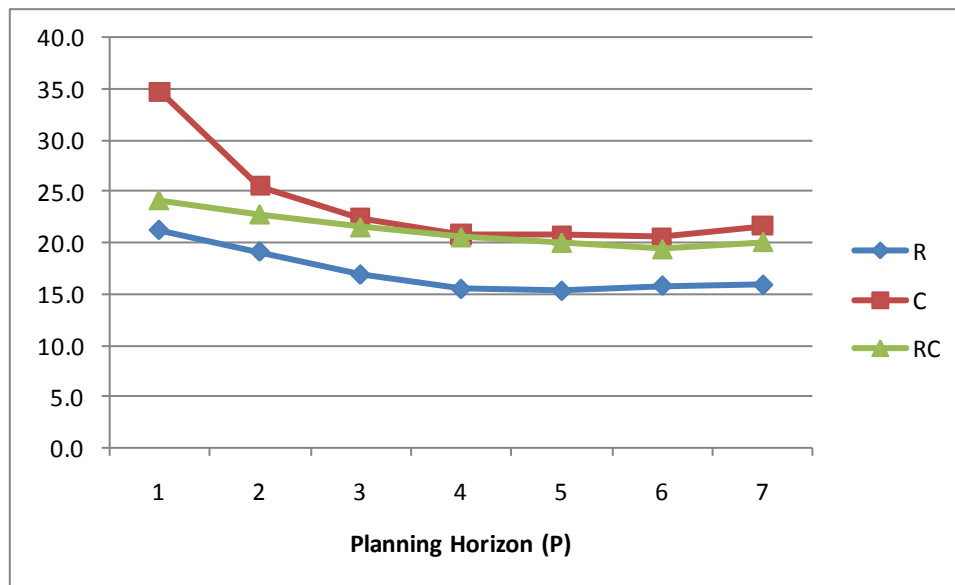


Figure 6.9: Average cost per routed customer (per problem type)

In terms of planning horizon  $P$ , in all problem types there is an appreciable decrease of the routing cost per customer up to a planning horizons of 4 periods. After that, and for all problem types, the unit routing cost reaches a plateau with a slight routing cost increase in the last two values of the planning horizon (6 and 7). This is also evident in Fig. 6.10 in terms of the grand average over all instances and problem types. The same Figure indicates that the total number of served customers does not exhibit significant variations among the different planning horizons.

In terms of problem type (R1, C1, RC1), it seems that all problem types present similar behavior regarding the unit cost change as the planning horizon increases from 1 to 7 periods; the C1 instances present the largest unit routing cost decrease.

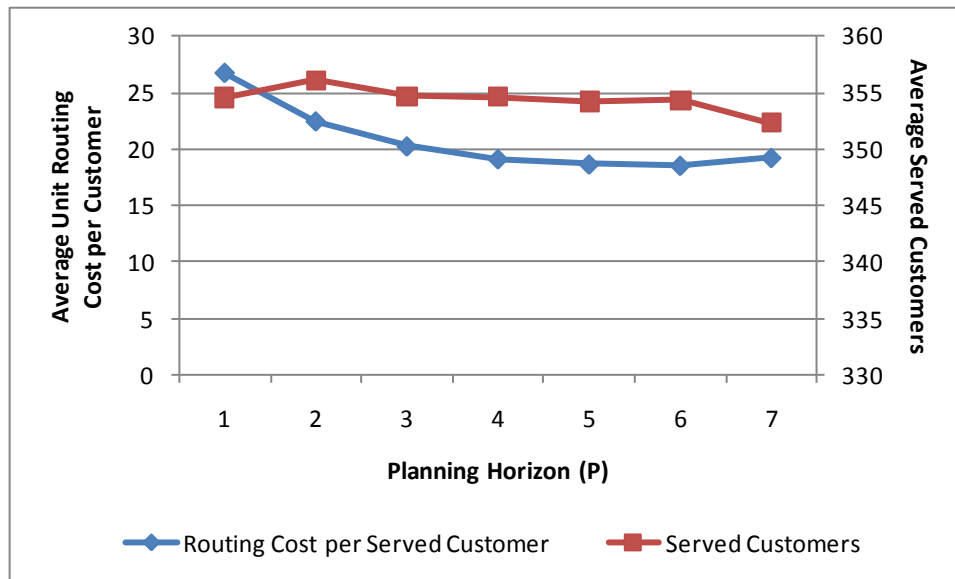


Figure 6.10: Routing cost per served customer and served customers per planning horizon (average over all instances and types)

The above observations are validated by the Analysis of Variance (ANOVA) conducted for factor  $P$  and for the cost per customer (output). Two analyses were conducted in order to verify the initial favorable effect of increasing  $P$ , followed by the settling of the output value beyond a certain  $P$  value: The first analysis concerned the effect of  $P$  over all 7 levels (planning horizons from 1 to 7), while the second examined its effect over the last 4 levels (planning horizons from 4 to 7). The analysis used all nine test instances, and the results are shown in Tables 6.15 and 6.16, respectively.

Table 6.15: ANOVA for factor  $P$  varying from 1 to 7 (7 levels) w.r.t. cost per served customer

Groups	Count	Sum	Average	Variance
1	9	240.511	26.723	40.000
2	9	202.511	22.501	9.494
3	9	182.895	20.322	8.613
4	9	170.828	18.981	7.450
5	9	168.514	18.724	7.946
6	9	167.306	18.590	4.843
7	9	173.285	19.254	9.058

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	477.638	6	79.606	<b>6.375</b>	0.000	<b>2.266</b>

<b>Within Groups</b>	699.234	56	12.486
<b>Total</b>	1176.872	62	

Table 6.16: ANOVA for factor  $P$  varying from 4 to 7 (4 levels) w.r.t. cost per served customer

Groups	Count	Sum	Average	Variance
<b>4</b>	9	170.83	18.98	7.45
<b>5</b>	9	168.51	18.72	7.95
<b>6</b>	9	167.31	18.59	4.84
<b>7</b>	9	173.28	19.25	9.06

ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
<b>Between Groups</b>	2.33	3	0.78	<b>0.11</b>	0.96	<b>2.90</b>
<b>Within Groups</b>	234.38	32	7.32			
<b>Total</b>	236.70	35				

Tables 6.15 and 6.16 show that the effect of  $P$  observed in Figs. 6.9 and 6.10 is statistically significant, that is, wider planning horizons succeed in improving routing costs. However this effect indeed reaches a plateau for  $P = 4, \dots, 7$ , in which there is no significant variation of the cost per customer (the F-value is lower than the F-critical value).

An additional analysis was conducted in order to test the effect of  $P$  on the number of customers served (planning horizons from 1 to 7). The results are presented in Table 6.17 and validate that there is no significant difference in the number of routed customers (F-value = 0.76 vs. critical F-value = 2.27).

Table 6.17: ANOVA Test for factor  $P$  (7 levels) w.r.t. Served Customers

Source of Variation	SS	df	MS	F	P-value	F crit
<b>Between Groups</b>	2338.16	6	389.69	<b>0.76</b>	0.60	<b>2.27</b>
<b>Within Groups</b>	28547.11	56	509.77			
<b>Total</b>	30885.27	62				



## Chapter 7: THE MPVRPTW WITH PRE-ASSIGNED CUSTOMERS

In this Chapter we address an interesting problem that stems from practice. Consider an environment in which a fleet of vehicles serves two types of customer orders over a time horizon:

- First, each vehicle should serve certain known customer orders pre-assigned to it, per period of the horizon. The sequence followed by a vehicle to serve these "inflexible" orders is not fixed. Furthermore, the sets of pre-assigned orders per vehicle vary, in general, from period to period of the horizon.
- Secondly, within this horizon, the fleet serves "flexible" customer orders that arrive dynamically and are characterized by a certain period window, and a certain time window.

The problem posed in this Chapter is to serve both the inflexible and the flexible customer orders with the minimum routing cost. This problem is solved on a rolling horizon basis in order to address the dynamics of arriving orders.

There are significant operational parameters to be considered in this environment, which make the problem both interesting and complex. These include the following:

- *Assignment of flexible customer orders:* As mentioned above, the flexible orders should be assigned to the vehicles serving the pre-assigned, inflexible customer orders.
- *Dynamic arriving process of flexible orders:* A flexible order that arrives in the current period, say  $p_c$ , may be served in a period window  $[\xi_i^s, \xi_i^e]$  where  $\xi_i^s > p_c$ . This implies that not all flexible orders that may be served in period  $p$  [ $p_c + 1 < p \leq P$ ] are known in the current period  $p_c$ . However, all flexible orders to be served in period  $p_c + 1$  are known in period  $p_c$ .

The main concept of our proposed approach is outlined below. As in Chapter 6, each flexible order  $i$ , for which  $\xi_i^e = p_c + 1$  must be assigned in period  $p_c + 1$ , while all other flexible orders with expiration period  $\xi_i^e > p_c + 1$  can be assigned within their respective period window interval  $[\xi_i^s, \xi_i^e]$ . Each pre-planned inflexible (or *mandatory*) customer order  $i^r$ , assigned to a specific vehicle  $r$ , may be considered as an order with period window of a single period, i.e.  $\xi_i^s = \xi_i^e$ , and, is also assigned to a specific vehicle within the period of service.

The assignment of flexible orders is performed in the most cost effective manner. Having assigned all known flexible orders within the planning horizon  $[p_c + 1, p_c + P]$ , the orders in the first  $M$  periods of the planning horizon  $P$  are implemented.

The importance of this special environment is discussed in the next Section. Note that the pre-assignment of inflexible customer orders necessitates modifications to the multi-period models presented in the previous Chapters. Furthermore, to solve the resulting problem, certain modifications are required in the proposed method, including the column generation scheme. These are discussed in the subsequent Sections.

## **7.1 PRACTICAL APPLICATIONS OF THE MPVRP WITH PRE-ASSIGNED CUSTOMERS**

The above planning problem is encountered in several supply chains. Typical examples include environments in which: (a) the planning (routing) process is performed in several batches throughout the day; in this case after each batch is scheduled, the related orders are assigned to certain vehicles, and (b) Fixed routes are predefined in an effort to estimate the expected workload in a, typically short-term planning horizon. (c) There are specific customers that demand a daily or periodic service (i.e. bank, grocery stores). Typically, such customers are pre-assigned to certain vehicles (e.g. that serve specific geographical areas). Additional customer orders that may become known are allocated taking under consideration these pre-assigned customers orders (inflexible orders).

All these environments are further discussed below.

### **Example 1: Strong dependence between routing and picking**

Consider an operational environment, in which the picking process is time consuming. Additionally, the customer orders arrive in batches, with the last batch arriving close to the start of delivery operations. In such a situation, the picking process (i.e. the collection of the items from the warehouse and their assignment to the vehicle loading zones), as well as the loading of the vehicles, start prior to obtaining the information on all customer orders to be delivered in the next scheduling period; this pre-emption is necessary in order to distribute evenly the picking and vehicle loading work, and avoid operational delays. For a certain batch of orders to be planned (flexible orders), the ones already assigned to vehicles are the inflexible ones.

Having pre-assigned customer orders to the vehicles, there is a possibility that not all flexible customer orders can be served within the same period (day). Thus, the planners need to select which orders to deliver during the next period and which to postpone for a later period. The customer orders that have not yet been assigned to vehicles can be considered as the flexible orders. Postponing some of the flexible orders necessitates to consider the MPVRP with pre-assigned customers.

**Example 2: Periodic service of significant customers**

This case is related to environments, in which certain customers require daily or periodic service. Usually these customers are considered as key accounts, which provide significant revenue to the distribution business. Characteristic examples include: (a) Express courier services that serve bank branches requiring daily service, as well as (b) distribution of groceries, fresh, or perishable goods to minimarkets or supermarkets, to which the products are delivered in set frequency (e.g. every 2 days). The current operational practice is to serve these customers by predefined routes in order to simplify the warehousing and distribution processes while providing increased service quality. These customers are considered as the inflexible ones since they need to be serviced with high priority. Apart from these orders, there are also additional customer orders that are concerned as 2<sup>nd</sup> priority orders. Based on each company's service level agreement, these orders may be served within a period window of consecutive days upon their arrival, allowing the planners to provide a more cost effective distribution planning.

**Example 3: Next day delivery mixed with micro-logistics operations**

This third case is related to operational environments that are characterized by a mix of customer service levels. Such environments are, among others, courier services. A typical courier network consists of several service centers, which are responsible for the distribution and collection of parcels and letters using a dedicated fleet. The main tasks of a service center can be summarized in (a) deliveries, (b) pickups, and (c) bulk product deliveries. Tasks (a) and (b) are the inflexible orders, which arrive typically overnight or during the early morning prior to the beginning of the mandatory service period. Tasks (c) are flexible, arrive daily but should be served within the next  $P$  periods (days) after arrival. A mix of 80% inflexible and 20% flexible calls is typical in many courier operations. Typical tasks belonging to the latter category (c) include several micrologistics activities, such as the distribution of high tech items, e.g. mobile phone sets, or internet kits, as well as bulk deliveries of advertising products and materials. For these tasks, the customers to be served should be informed at least

one period prior to the actual service delivery. Due to the large volume of batch arrivals of orders, and considering the nature of the product been delivered, it is both inefficient and unnecessary to serve all micrologistics orders within a single period. Thus, the flexible orders must be assigned to the periods of the planning horizon prior to knowing the inflexible orders, in order to be able to inform the customers regarding the day and time of the scheduled visit. In order to do this, one uses typical routes that the fleet vehicles are likely to travel during each period of the planning horizon. These routes may vary or be the same within the horizon, or may follow seasonal patterns.

#### **Example 4: Dealing with routing uncertainty in field service environments**

Another operational environment of relevance to the problem considered here concerns maintenance/repair services that are delivered on-site. In this environment, a group of repair persons provide services on location (e.g. appliance, or equipment maintenance).

The inflexible calls typically are repair tasks that (a) are pre-scheduled in a certain day (i.e. for preventive maintenance) or (b) they need immediate attention (and generate increased revenue), while flexible calls are the ones that may not need to be addressed immediately but within a selected period window set by the customer. Pre-assignment of inflexible orders to specific vehicles (or driver) may be performed based on the equipment or skills required.

In such operational environments, customer service is typically problematic, forcing customers to wait for unspecified time within the promise day of service delivery. The main reason for this difficulty is that service planners have no prior knowledge of the total picture of the pending tasks, as well as of the dependencies among them (priorities, adjoined orders, etc). The decisions are mostly based on experience and typically each day is taken as independent from the others, without taking into account the characteristics of the demand.

## **7.2 MODIFICATIONS FOR THE MULTI-PERIOD ROUTING PROBLEM WITH PRE-ASSIGNED CUSTOMERS**

### **7.2.1 MODIFICATIONS IN THE MATHEMATICAL MODEL**

In this Section, we modify the mathematical formulation presented in Chapter 3 and Section 6.3 in order to take under consideration the existence of pre-assigned customers. Hereafter for simplicity, and without loss of generality, the current period (planning period) is set to  $p_c = 0$ , and the planning horizon is  $[1, P]$ .

Let  $N^m$  be the set of all inflexible customer orders, and  $N^f$  be the set of all known flexible orders, thus,  $N^m \cup N^f = N$ . Each inflexible customer  $i^r$  is assigned a period window  $[\xi_i^s, \xi_i^e] = [\rho_i, \rho_i]$  where  $\rho_i$  is the period within which customer  $i^r$  should be served. In addition, each inflexible order  $i^r$  is assigned to the route of the corresponding vehicle  $r$  from the available vehicle set  $K_p = \{k_p^1, \dots, k_p^r, \dots, k_p^{|K_p|}\}$ .

In order to force inflexible customer orders to be served within a certain period by a designated vehicle, while the flexible orders can be served once within their assigned period window by any vehicle, Constraint (3.2) is modified into two separate constraints, one for the inflexible and one for the flexible customer orders, as follows:

$$\sum_{j \in W} x_{ij\rho_i k_{\rho_i}^r} = 1 \quad \forall i^r \in N_m \quad (7.1)$$

$$\sum_{p \in I_i} \sum_{k \in K_p} \sum_{j \in W} x_{ijpk} = 1 \quad \forall i \in N_f \quad (7.2)$$

Constraint (7.1) specifies that each inflexible customer order should be served during the required (single) period and by the designated vehicle, while Constraint (7.2) specifies that each flexible customer will be served only once by any available vehicle and in any period  $p$  within its period window  $I_i$ . Note that in the aforementioned business setting, it is possible to have vehicles (routes) without assigned inflexible orders. The proposed modifications are capable of handling such a situation, since these vehicles will contain only flexible customers considered by Constraint (7.2).

## 7.2.2 MODIFICATIONS OF THE COLUMN GENERATION METHOD

In order to consider the assignment of inflexible customer orders to vehicles, several modifications are required in the column generation method presented in Chapters 4, 5, and 6. Similar modifications were first discussed and presented by Ninikas and Minis (2011) for the Vehicle Routing Problem with Dynamic Pickups (VRPDP).

Given the initial assignment of inflexible customer orders, the initial columns (routes) of the Restricted Master Problem (RMP) should, at least, contain these inflexible orders. Note that the RMP requires an initial feasible solution in order to be solved, and since the inflexible customer orders are pre-assigned to vehicles (i.e. they are feasible only within the pre-assigned vehicles and periods), such routes need to be provided. Recall that in the cases without inflexible customer orders (discussed in Chapters 3 to 6) the following hold:

- (a) For the case of unlimited vehicle fleet, the initialization may be performed using the unit routes (i.e. routes that visit only one customer and return to the depot, since such routes are feasible, although trivial).
- (b) For the case of limited fleet, but without inflexible customers (see Chapter 3), the initialization may be performed through heuristics that provide an initial feasible solution to the underlying problem, in which all orders are served and the fleet size within each period is respected.
- (c) For the case of limited fleet, when using the modified objective function described in Chapter 6, the initial solution can be provided either using a heuristic (as described in case *b*) or by using the virtual unit routes (columns) that are associated with the unserved customers penalty costs.

All above initialization schemes cannot be utilized for the current case, since pre-assigned customer orders should be taken under consideration along their assignment to certain vehicles. We initialize the method using a trivial solution that includes the routes that contain only the inflexible orders assigned to the vehicles of each period, while the flexible orders are considered as unserved. The latter are treated as unserved in this initial solution, by using the virtual unit columns that correspond to leaving these customers unserved (see Section 6.3).

Since each available vehicle is associated with a set of inflexible customer orders, considering a single subproblem per period is not appropriate. The reason is that the inflexible customer orders assigned to each vehicle need to be treated in a way that prevents creating columns (routes) that contain inflexible orders pre-assigned to different vehicles. In order to address this issue, a straightforward modification is utilized that considers one subproblem per each period-route combination. For a relevant approach in the case of the multiple depot MPVRP, see Tricoire (2007).

### **Modifications to the Subproblem**

In the current case, each subproblem is associated with a period  $p$ , a vehicle  $k_p^r$ , a set of inflexible customers  $N_m^r = \{i^r \in N_m, p = \rho_i\}$ , and a set of flexible customers  $N_f^r = \{i \in N_f, p \in [\xi_i^s, \xi_i^e]\}$ . In order to generate feasible columns (routes), the following modifications should be incorporated into the method presented in Section 3.3 and Chapter 4:

- Each generated column (route) should include all inflexible customers of set  $N_m^r$ , i.e. the set corresponding to the appropriate vehicle. Thus, a label  $L_{\delta i}$  related to this route is not

allowed to be extended to the depot if not all customers within  $N_m^r$  have already been served.

- Additionally, each generated column (route) should not include any inflexible customer not associated with vehicle  $r$ . This is addressed by including in the subproblem corresponding to vehicle  $k_p^r$  only the corresponding feasible inflexible customers (i.e.  $N_m^r$ ).
- In case a mandatory customer from set  $N_m^r$  becomes *unreachable* within a label  $L_{\delta i}$ , this label is discarded and it is not extended further, since it cannot include all inflexible customers.

### Required modifications for the dominance criteria

The existence of inflexible customers within each route requires the enhancement of the dominance criteria in order to consider the number of inflexible customers that have been served by the associated partial route  $\delta$  of a label  $L_{\delta i}$  when it is compared to another label  $L_{\delta i}$ . A straightforward modification would allow to compare for dominance two different labels ending at the same node (customer) only if the associated partial paths have served the same inflexible customers. Doing so, the dominance criteria would be applied only to a very limited number of labels, leading to an impractically high number of partial routes to be extended, and to prohibitive computational times.

In order to overcome this issue, we enhance the labels by adding a cost factor  $\bar{c}_{\delta i}$  (*equilibrium cost*) that represents an upper bound (worst case) of the total modified cost required to visit all inflexible customers not yet served following partial route  $\delta$ . This is done by taking into account only connecting arcs which are feasible (and, thus, not included in unreachable vector  $R_{\delta i}$ ); that is,

$$\bar{c}_{\delta i} = \sum_{i^r \in N_m^r \setminus \{i^r: i^r \notin \delta\}} \left( \max_{h \in N \cup \{0\} \setminus \{h: R_{\delta i}^h = 0\}} (c'_{hi^r}) + \max_{j \in N \cup \{n+1\} \setminus \{j: R_{\delta i}^j = 0\}} (c'_{i^r j}) \right) \quad (7.3)$$

where  $c'_{ij}$  is the modified cost associated with arc  $(i, j)$ , while  $R_{\delta i}^h$  is the element of the unreachable vector  $R_{\delta i}$  within the partial route  $\delta$  that is associated to node (customer)  $h$  (note that when an element  $R_{\delta i}^h$  is equal to 1, then order  $h$  has been already visited or cannot be visited by partial path  $\delta$ , while  $R_{\delta i}^h = 0$  denotes that customer  $h$  has not yet been included in the partial route  $\delta$ ).

With this modification, label  $L_{\delta i}$  becomes  $L_{\delta i} = [\bar{c}_{\delta i}, \bar{c}_{\delta i}, t_{\delta i}, d_{\delta i}, R_{\delta i}]$ , and the related dominance criterion used in the procedure of Section 3.3 is given by:

$$\bar{c}_{\delta' i} \leq \bar{c}_{\delta'' i} \quad (7.4)$$

Note that the additional dominance criterion does not violate optimality when the associated ESPPTWCC is solved within a full B&P framework, since it eliminates labels that lead to worst routes with respect to the reduced cost.

### 7.3 EXPERIMENTAL INVESTIGATION OF ROLLING HORIZON WITH PRE-ASSIGNED CUSTOMERS

The experimental investigation of the present problem is similar in scope to the one described in Section 6.4; that is, it seeks to determine the significance of the length of planning horizon  $P$  in settings in which a part of the known customers has already been pre-assigned to certain vehicles. We investigate the *dynamic* rolling horizon routing problem, in which customers become known one period prior to the opening of their period window.

The experimental testbed was constructed as follows:

- The test instances described in Section 6.4. were used to define the customer geographical distribution, as well as the customer parameters (time window, demand, etc)
- We considered three different period window patterns, that is 3, 5 and 7, in order to simulate various degrees of customer period flexibility in this multi period setting.
- From the customer set of each instance, we randomly selected 180 customers as the inflexible ones. We distributed these customers in subsets of 6 to the 30 periods of the long-term horizon (one set per period). All inflexible customers were assigned the widest possible time window (equal to the maximum available routing time per period) in order to avoid routing infeasibilities and, also, to not limit the assignment of flexible orders to the vehicles due to limited resources (e.g. increased waiting times due to the time windows of the inflexible customer orders). Finally, to serve the customers of each period we used two vehicles with maximum service limit of 3 inflexible customers orders per vehicle; thus we developed two routes per period.
- In addition to the inflexible customers, 6 dynamic customer orders arrived per period (e.g. 180 flexible customer orders in total). To achieve a smooth initial transition, we employed

a “warm start” by considering dynamic orders from 3 past periods prior to period 1 (e.g. in the first period there are 18 dynamic customers that are considered for planning).

- The output parameters studied concern the number of flexible customer orders served, and the *additional* cost per flexible customer order. This additional cost is derived by subtracting from the overall routing cost of each period, the original routing cost of the inflexible orders in the same period. To obtain the *routing cost ratio*, this additional cost is divided by the number of flexible orders served in this period.

### 7.3.1 EXPERIMENTAL RESULTS

Table 7.1 presents the results per problem type, period window pattern and planning horizon. The values presented in this Table are (a) the average number of served flexible customers over the entire horizon, and b) the average cost ratio (or unit cost) ; these averages were obtained over the three test instances of the corresponding problem type. (e.g. for R1 the values corresponding to the instances R103, R106 and R109 have been averaged). Appendix E presents the results per instance.

Table 7.1: Average experimental results per problem type, period window pattern, and planning horizon

P	Problem Type					
	R1		C1		RC1	
	Served Customers	Unit Cost per Customer	Served Customers	Unit Cost per Customer	Served Customers	Unit Cost per Customer
Period Window Pattern 3						
1	176.0	13.8	180.0	19.6	162.3	12.0
2	177.0	8.2	180.0	9.5	164.3	11.0
3	177.0	7.0	180.0	8.0	163.0	10.2
Period Window Pattern 5						
1	177.3	13.1	180.0	19.5	174.7	11.9
2	177.3	7.1	180.0	8.3	176.3	9.5
3	177.3	4.4	180.0	5.2	176.3	8.2
4	177.3	4.1	180.0	4.1	176.0	5.7
5	177.3	4.1	180.0	4.2	175.7	6.4
Period Window Pattern 7						
1	177.3	13.0	180.0	19.5	174.3	11.4
2	177.3	7.0	180.0	8.1	176.3	9.3
3	177.3	4.1	180.0	4.6	176.7	6.8
4	177.3	3.7	180.0	3.8	176.3	4.6
5	177.3	3.2	180.0	3.2	175.3	4.0
6	177.0	3.1	179.7	3.0	175.3	4.0
7	177.0	2.9	179.7	3.0	175.3	3.8

For the three period window patterns, Figures 7.1, 7.2 and 7.3 display the average unit cost per routed dynamic customer order for all planning horizon values, and for the three customer distribution types.

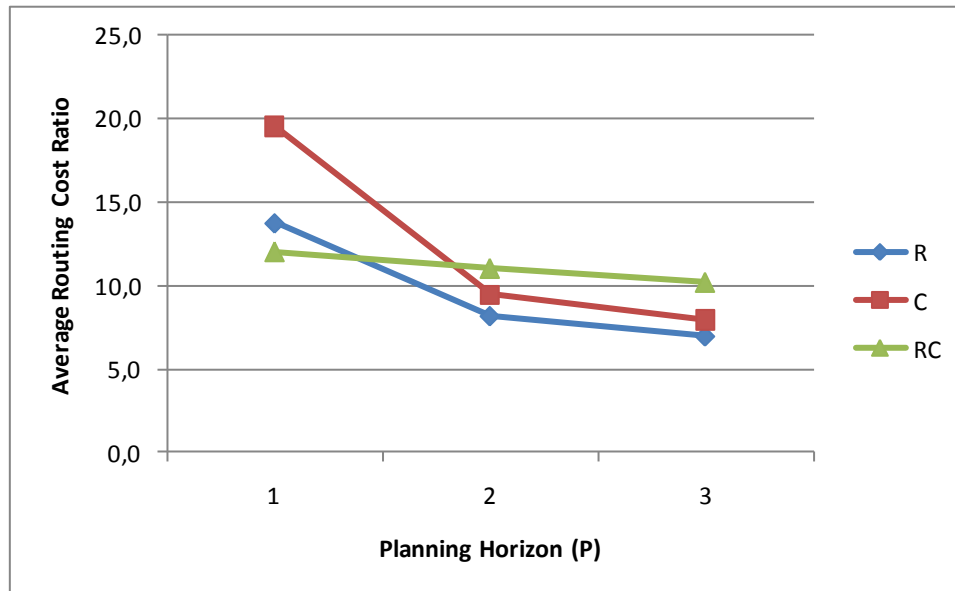


Figure 7.1: Average routing cost ratio per problem type (Pattern 3)

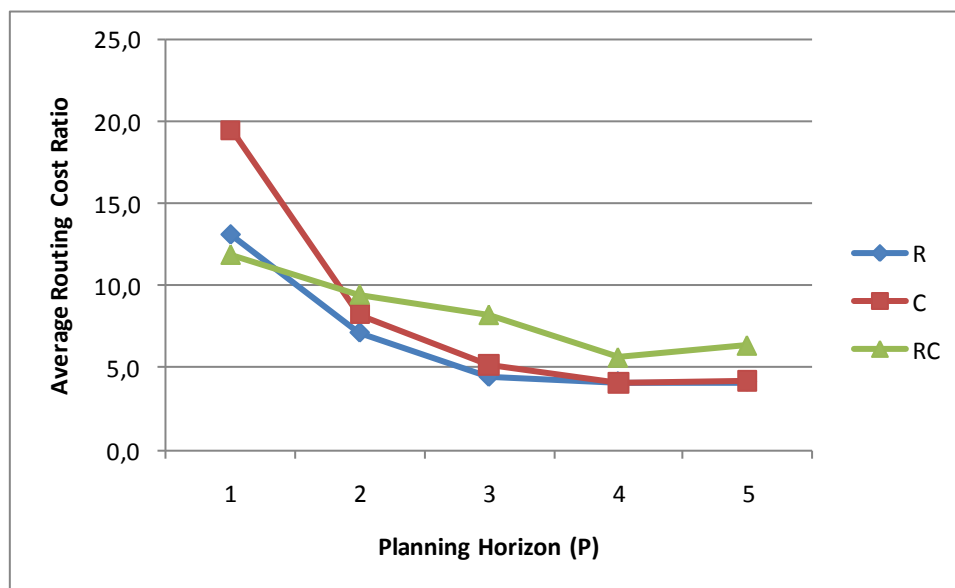


Figure 7.2: Average routing cost ratio per problem type (Pattern 5)

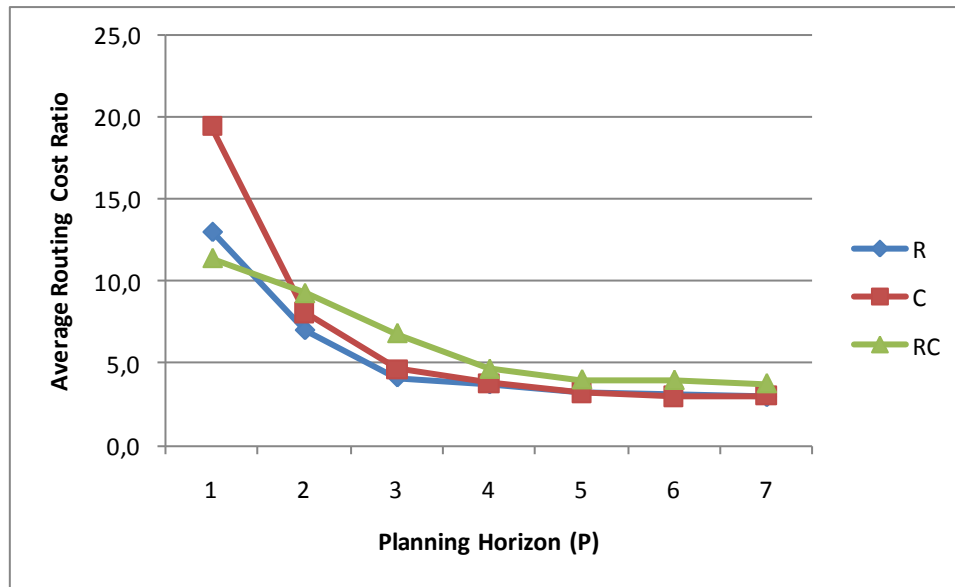


Figure 7.3: Average routing cost ratio per problem type (Pattern 7)

In terms of the planning horizon  $P$ , in all problem types and, specifically, in the period window patterns 5 and 7, the routing cost per customer decreases significantly in the initial range of  $P$  values. This decreasing trend reaches a plateau beyond a certain value of  $P$  (e.g.  $P = 4$  for Pattern 7). In terms of problem type (R1, C1, RC1), it seems that the decrease of the cost ratio is more pronounced for the R1 and C1 problems, while the RC1 problem type presents a more limited decreasing trend.

Figures 7.4, 7.5 and 7.6 present the grand average over all instances and problem types per period window pattern. These Figures validate the significant decrease of the routing cost ratio with increasing values of the planning horizon. Furthermore, they indicate that the total number of served customers is not affected significantly by the planning horizon value.

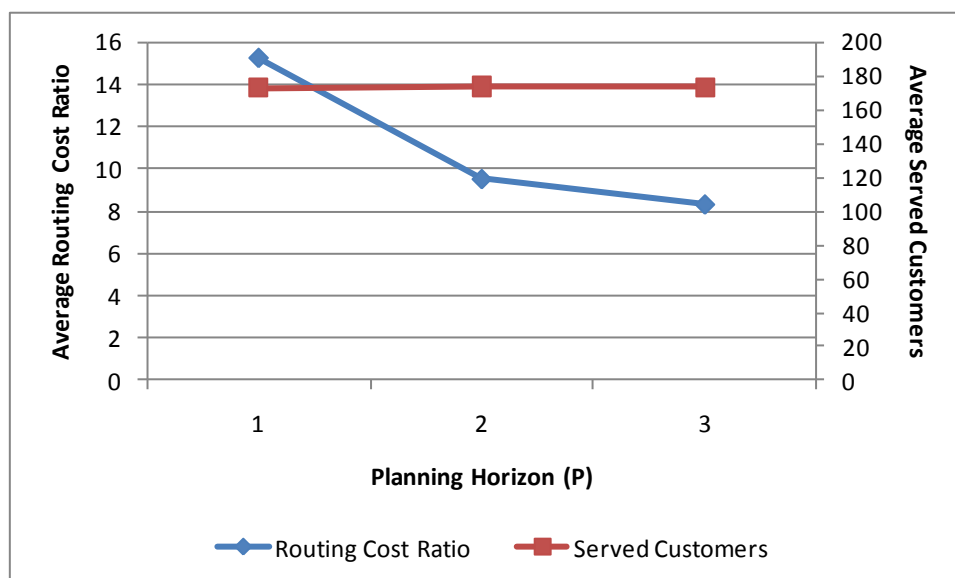


Figure 7.4: Routing cost ratio and served customers per planning horizon (average over all instances) – Pattern 3

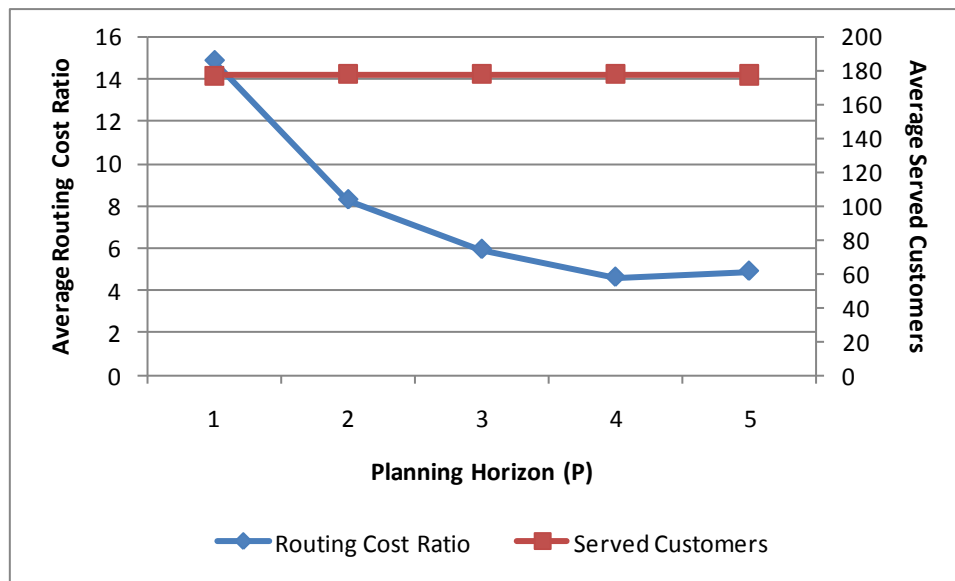


Figure 7.5: Routing cost ratio and served customers per planning horizon (average over all instances) – Pattern 5

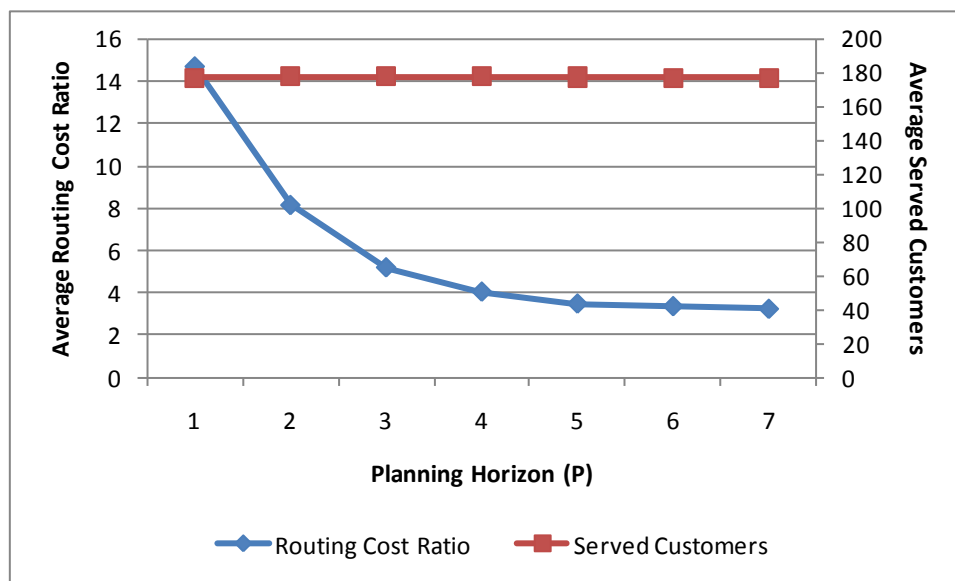


Figure 7.6: Routing cost ratio and served customers per planning horizon (average over all instances) – Pattern 7

### Statistical validation of the results

The above observations are validated by a paired difference t-test, which was conducted for the factor  $P$  (input) and the routing cost ratio (output). For the results of each period window pattern (3, 5 and 7), two paired difference t-tests were conducted.

- The first concerns the comparison of the routing cost ratio between the levels  $P = 1$  and that corresponding to the medium range of  $P$  (e.g.  $P = 2$  for pattern 3,  $P = 3$  for pattern 5, and  $P = 4$  for pattern 7).

- The second test concerns the comparison of the routing cost ratio between the level corresponding to the medium range of  $P$  and the highest  $P$  level (e.g. 3, 5, 7 for patterns 3, 5 and 7, respectively).

The hypothesis tested is the following:

$$H_0: \text{Average difference of Routing Cost Ratio} = 0$$

$$H_1: \text{Average difference of Routing Cost Ratio} > 0$$

Since we are testing only for the difference of the cost ratio been higher than zero, the one-tail t-value is relevant. The degrees of freedom for all samples are equal to 8 (since each data set contains nine samples). Based on a 95% confidence level, the related t-value is:  $t_{0.95,8}^{one\ tail} = 1.86$ . Table 7.2 presents the results of the paired difference t-tests.

Table 7.2: Paired difference t-test analysis for factor  $P$  and routing cost ratio

<b>Pattern 3</b>	<b>Routing Cost</b>					
	<b>P=1</b>	<b>P=2</b>	<b>Dif.</b>	<b>P=2</b>	<b>P=3</b>	<b>Dif.</b>
<b>Mean</b>	15.12	9.60	5.53	9.60	8.42	1.18
<b>Variance</b>	12.53	3.43	16.10	3.43	3.79	0.16
<b>t Stat</b>	<b>4.13</b>			<b>8.75</b>		
<b>Pattern 5</b>	<b>P=1</b>	<b>P=3</b>	<b>Dif.</b>	<b>P=3</b>	<b>P=5</b>	<b>Dif.</b>
<b>Mean</b>	14.82	5.95	8.88	5.95	4.92	1.03
<b>Variance</b>	13.89	5.44	22.14	5.44	3.04	0.76
<b>t Stat</b>	<b>5.66</b>			<b>3.55</b>		
<b>Pattern 7</b>	<b>P=1</b>	<b>P=4</b>	<b>Dif.</b>	<b>P=4</b>	<b>P=7</b>	<b>Dif.</b>
<b>Mean</b>	14.64	4.05	10.59	4.05	3.24	0.81
<b>Variance</b>	14.72	0.88	16.22	0.88	0.70	0.04
<b>t Stat</b>	<b>7.89</b>			<b>12.78</b>		

For both t-tests the null hypothesis is rejected (the t-Stat value is larger than the  $t_{0.95,8}^{one\ tail}$  value in all paired differences). Thus, the routing cost ratio decreases significantly as the planning horizon widens in all tested period window patterns.

Using similar hypotheses, the paired difference t-test w.r.t. the number of served customers validates that  $P$  does not have a significant effect on the former.

### Indicative experimental results for a case with a large number of unserved customers

We have also investigated cases in which only a limited number of flexible customers can be serviced. The scope of this investigation was to assess the proposed approach under such

extreme situations. To do so, we utilized pattern 7 and increased the assigned service time of each flexible customer order by 100% (w.r.t. the service time of the test instances discussed in Section 6.4). All other parameters used in the previous setting remained intact. Figure 7.7 presents the grand average over all instances per period window pattern.



Figure 7.7: Routing cost ratio and served customers per planning horizon (average over all instances) – Pattern 7

In this case, the number of served flexible orders is limited compared to the total of 180 flexible orders. Additionally, the routing cost ratio does not indicate significant variation. This is possibly due to the high ratio of unserved customer orders, which increases significantly the penalty costs, and suppresses any routing cost variation.

To investigate this further, we tested different penalty cost functions in addition to the linear one ( $\gamma = 5$ ). Thus we experimented with  $\gamma = 1$  (flat),  $\gamma = 2$  (step), and  $\gamma = 3$  (square). The results for the average routing cost ratio and the average number of served customers are presented in Figures 7.8 and 7.9, respectively.

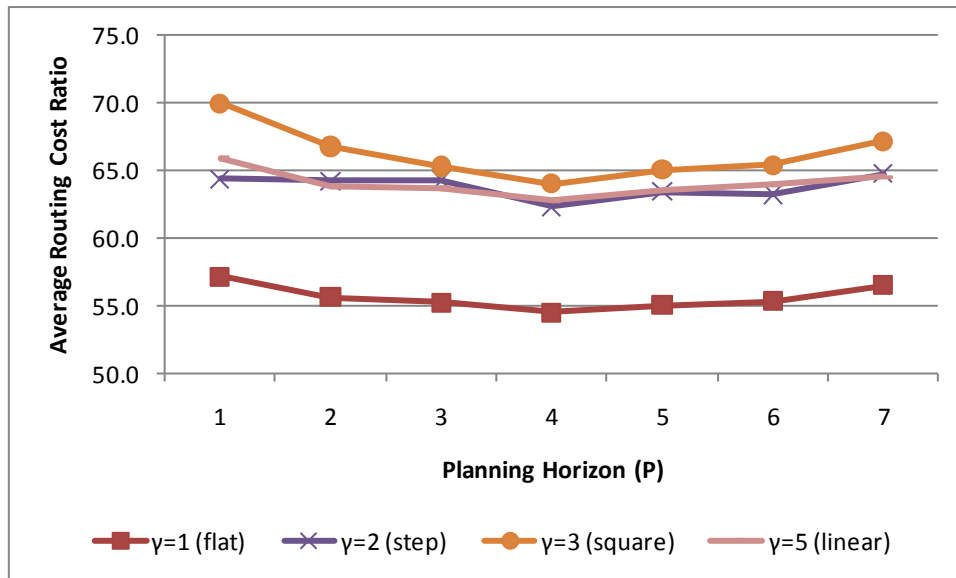


Figure 7.8: Average routing cost ratio per planning horizon and penalty function (average over all instances) – Pattern 7

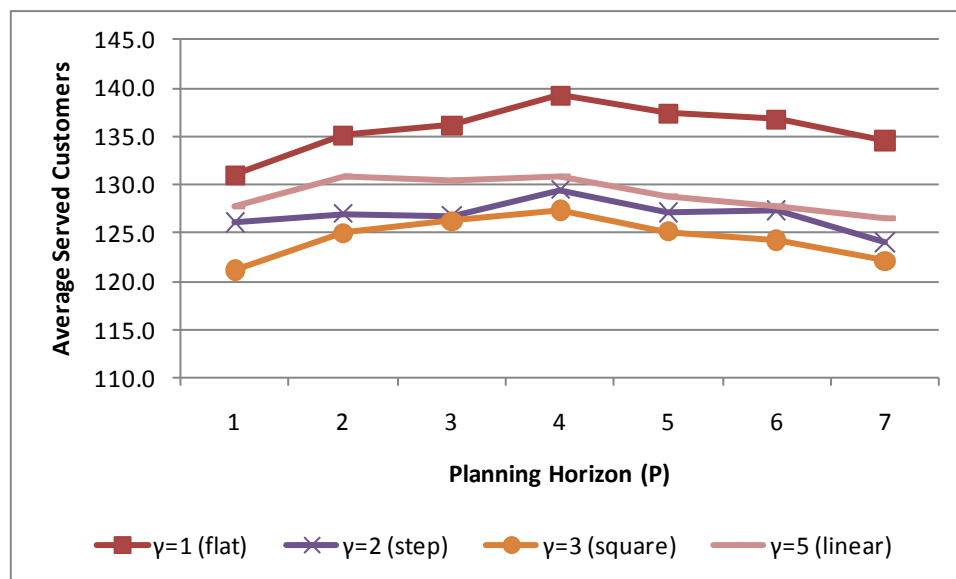


Figure 7.9: Average served customers per planning horizon and penalty function (average over all instances) – Pattern 7

The flat penalty function ( $\gamma = 1$ ), in which all flexible customer orders are given the same penalty cost, seems to provide improved results (in both routing cost and served customers) in comparison with the other penalty functions. This can be explained by the fact that (a) the flat penalty function does not interfere with the routing procedure by prioritizing expiring customers, and (b) many flexible customers will remain unserved anyway.

As far as the planning horizon is concerned, the moderate planning horizon values seem to provide improved results in served customers per period, and slight improvements with respect to routing costs.

## Chapter 8: CONCLUSIONS AND FUTURE RESEARCH

### 8.1 CONCLUSIONS

In this dissertation we studied the Multi-Period Vehicle Routing Problem with Time Windows (MPVRPTW). We provided the mathematical formulation of the MPVRPTW in conjunction with its remodeling into a framework amenable to column generation.

#### Efficient lower bounds for the relaxed MPVRPTW

Based on the insights in the column generation method, we developed two different exact strategies to provide lower bounds to the linear relaxation of the MPVRPTW:

- The cloning strategy transfers feasible routes generated by one subproblem to the other subproblems of the column generation scheme. This strategy seeks computational savings by avoiding to solve explicitly the subproblem of each period
- The unified strategy solves a single (unified) subproblem that considers all periods of the planning horizon. Period feasibility of each generated route is checked within this subproblem.

These strategies take advantage of the special structure of the multi-period problem, such as the flexibility of customers to be routed in different periods, and the existence of routes that may be assigned to multiple periods. We studied the efficiency of the proposed methods (w.r.t. computational savings) against the classical adaptation of the column generation method to the multi-period setting, as well as against its parallel implementation.

The two alternative methods (cloning and unified) succeed in reducing the computational time, compared to the above reference methods, with the exception of the clustered instances in the parallel implementation. Specifically, for the random and the clustered instances, the cloning method exhibits the best performance with time savings of about 50% with respect to the classical method. For the mixed test instances, the unified method appears to be the most efficient, resulting to the highest time savings (also about 50%). While for all problem sets, the parallel and the cloning methods appear to be the most efficient.

Additionally, as expected, for narrow period windows the proposed algorithms do not provide substantial efficiency gains, due to the limited customer flexibility and the limited similarity between the subproblems. On the contrary, significant efficiency gains appear in wider period

windows (patterns 6 to 9), which are the most computationally expensive instances. The unified method exhibits the most diverse behavior regarding the period window patterns: It presents the least efficient results for narrow period windows (with even 2 times greater computational times for pattern 1 compared to the classical reference method); however, for the wider period window patterns, it outperforms all other algorithms, succeeding in a 62% reduction for pattern 9 with respect to the classical method.

The efficiency of all methods seems to be more evident in clustered instances where there are computational savings even for moderate period windows. For the mixed instances, the parallel implementation presents more consistent performance compared to the other methods, while for the random instances efficiency gains are realized only for wide period windows.

#### Integer Solutions for the MPVRPTW

Integer solutions for the MPVRPTW are provided through a proposed branch-and-price implementation that is relevant to the multi-period setting. Two different strategies for exploring the branch-and-price tree have been discussed and tested: (a) The classical one, in which two branches are generated after each fractional solution, and (b) a slight modification that considers the multi-period characteristics of the problem by creating  $P + 1$  branches. Additionally, a simple pruning heuristic is proposed in order to accelerate the integer solution procedure. This heuristic stops the extension of the branch-and-price tree for not “promising” branches, for which the lower bound is close to the best known global upper bound. Thus, it is able to provide near-optimal results, also in instances with wide period flexibility of customer orders.

Based on the previous results, we selected to employ the cloning method in the B&P scheme and compared the efficiency of the latter against the B&P scheme that uses the classical reference method. For the cases for which an integer solution was obtained (within the computational time limit), the cloning method results in significant gains in determining the optimal (or a suboptimal) integer solution, especially as the width of the period window increases. For cases solved to optimality by B&P, the efficiency of the cloning method is moderated. This is attributed to the fact that the savings, stemming from determining the lower bound, are moderated by the other B&P operations, such as the generation of the B&P nodes.

To study the proposed pruning heuristic we focused on (a) those instances for which B&P determined the integer optimal solution, and (b) the instances for which an integer solution was found, but its optimality was not verified due to reaching the imposed time limit. The resulting cost deviation was found to be limited, below 1%, while the computational time has been reduced on the average by about 84% (category a) and 98% (category b). The significant benefits in computational times and the very limited deviation in the cost of the final solution indicated that the B&B heuristic is a very attractive alternative for large practical cases.

### Rolling horizon routing

Having set the foundations of the addressing the generic MPVRPTW, we focused on problems solved over long-term horizons. For these cases we proposed a rolling horizon framework and studied two arrival patterns of customer orders: (a) the quasi-static, and (b) the dynamic MPVRPTW. Considering the quasi-static case, we proposed and discussed three theoretical statements concerning the implementation horizon ( $M$ ) and the planning horizon ( $P$ ), thus, establishing the principles of applying the proposed framework and methods. Specifically, it was established that:

- The monolithic solution of the full multi-period routing problem (for the  $S$  period horizon) is always lower or equal to the final implemented solution obtained by any rolling horizon scheme with planning horizon of  $P < S$  periods,
- The overall routing cost provided by a rolling horizon scheme with planning horizon of  $P$  periods is not necessarily lower than or equal to the overall routing cost provided by a rolling horizon scheme with planning horizon of  $P'$  periods, where  $P' < P < S$ , and
- Using a rolling horizon scheme with planning horizon  $P > 1$ , it is not guaranteed that  $M = 1$  (i.e. implementing only the first period of the planning horizon) will always lead to a lower cost value compared to alternative implementation horizons with  $M > 1$ .

In order to address the MPVRPTW within a rolling horizon framework, we modified the model of the problem to take under consideration the case in which not all customer orders can be served within the planning horizon due to resource limitations. The problem is dealt through introducing penalty functions for the unserved customers, thus, balancing routing efficiency with the number of served customers within the long-term horizon.

Significant experimental investigation was performed considering both arrival patterns of customer orders. For the quasi-static case, longer planning horizons result in lower routing costs, validating the appropriateness and the efficiency of the proposed methods. In terms of

the implementation horizon  $M$ , it is clear that  $M = 1$  results in higher (or equal) number of routed customers in all cases, and, also, in lower routing cost per customer in almost all cases, except in the cases where  $M = 2$  resulted in a much lower number of served customers. These experimental results follow the same pattern for different time windows, as well as for different geographical distributions of customer orders. The above conclusions were also validated through appropriate statistical analysis.

For the dynamic rolling horizon routing case, and in all problem types, there is an appreciable decrease of the routing cost per customer up to a planning horizons of 4 periods. After that, the unit routing cost reaches a plateau with a slight routing cost increase in the last two values of the planning horizon (6 and 7). Also, the total number of served customers does not exhibit significant variations among the different planning horizons.

All problem types (random, mixed, clustered) present similar behavior regarding the unit routing cost, as the planning horizon increases from 1 to 7 periods; the clustered instances present the largest unit routing cost decrease. The experimental results also indicated that there is no significant variation in the number of routed customers.

#### The MPVRPTW with pre-assigned customers

The MPVRPTW with pre-assigned customers is related to environments in which inflexible and flexible customer orders co-exist. For this case, we proposed the required modifications in both the MPVRPTW model and the solution approach (column generation). We considered the dynamic arrival pattern of customer orders and tested three different period window patterns (3, 5 and 7). Extensive experimental investigation indicated that significant cost savings may be achieved by considering wider planning horizons in the planning process. In terms of the planning horizon  $P$ , in all problem types and, specifically, in the period window patterns 5 and 7, the routing cost per customer decreases significantly in the initial range of  $P$  values. This decreasing trend reaches a plateau beyond a certain value of  $P$  (e.g  $P = 4$  for Pattern 7). In terms of problem type, it seems that the decrease of the routing cost ratio is more pronounced for the random and the clustered instances, while the mixed instances present a more limited decreasing trend. Furthermore, the total number of served customers is not affected significantly by the planning horizon value.

For cases in which only a limited number of flexible customers can be served, we showed that the “flat” penalty function seems to provide improved results in both routing cost and served customers.

## 8.2 FUTURE RESEARCH

In order to accelerate the exact column generation method for the MPVRPTW, additional techniques that have already been proposed in the literature for vehicle routing or other problems, such as stabilization of the dual variables, constraints aggregation or cutting planes (k-path cuts, subset-row inequalities), can be studied.

Furthermore, intelligent heuristics, metaheuristics or hybrid approaches can be employed and compared to the methods proposed in this dissertation. For example, current research is focusing on hybrid approaches, in which column generation utilizes a heuristic or a metaheuristic (such as the tabu search) to solve the subproblems. These approaches may be promising in addressing large problem instances relevant to practical applications.

Beyond alternative or new methods to address the MPVRPTW, an interesting research direction is to consider other relevant problems, including

- The fully dynamic MPVRPTW, in which customer orders may arrive and be served during the execution of service to the customers (while vehicles are en-route). Column generation with the appropriate modifications may be utilized and tested for different planning horizons and penalty cost functions.
- The development of a stochastic model that concerns historical data and is able to either forecast demands or identify geographical areas of high demand density. Such a model may be combined with the MPVRPTW with pre-assigned customers and used in environments in which flexible orders are known in advance, and inflexible ones become known just prior to, or during, delivery.

In multi-period settings, the customer service level as defined by the actual period of service within the period window of each customer order, may be a significant operational quality indicator. Servicing the customer as early as possible within its period window may lead to increased customer satisfaction, while, on the other hand, it may increase operational costs. Enhancements of the MPVRPTW in order to balance service level and operational cost, may provide interesting results. A related noteworthy case concerns the incorporation of soft period windows to deal with the case of the limited resources and to minimize the number of unserved customer orders. In addition, the use of different penalties, not only based on expiration period, but also on the type of the customer, may provide a tool for prioritizing certain customers (i.e. key accounts) that are considered more important by a managerial / marketing or sales point of view.

Finally, in order to evaluate practical aspects of the current research, an initial case study in a Greek major courier service provider has been implemented and reported in Athanasopoulos and Minis (2011). This initial study has shown encouraging results. However, further investigation of the practical implications of the proposed methods is encouraged as part of future research.

## REFERENCES

- Agarwal, Y., Mathur, K., Salkin, H.M. (1989). A set-partitioning-based algorithm for the vehicle routing problem. *Networks*, 19, 731-750.
- Andreatta, G., Lulli, G. (2008). A multi-period TSP with stochastic regular and urgent demands. *European Journal of Operational Research*, 185, 122-132.
- 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, 106–119.
- Angelelli, E., Savelsbergh, M.W.P., Speranza, M.G. (2007). Competitive analysis of a dispatch policy for a dynamic multi-period routing problem. *Operations Research Letters*, 35(6), 713-721.
- Angelelli, E., Speranza, M.G., (2002). The periodic vehicle routing problem with intermediate facilities, *European Journal of Operational Research*, 137, 233-47.
- Athanasopoulos, T., Minis, I. (2011). Multi-period routing in Hybrid Courier Operations. In: Minis, I., Zaimpekis, V., Dounias, G., Ampazis, N. (Eds.). *Supply Chain Optimization, Design & Management: Advances & Intelligent Methods*, IGI Global Publications.
- Baldacci, R., Bartolini, E., Mingozzi, A., Valletta, A. (2011). An Exact Algorithm for the Period Routing Problem. *Operations Research*, 59, 228–241.
- 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, 261–300.
- Bard, J., Huang, L., Jaillet, P., Dror, M. (1998). A Decomposition Approach to the Inventory Routing Problem with Satellite Facilities. *Transportation Science*, 32, 189-203.
- 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), 316-329.
- Beasley, J., Christofides, N. (1989). An algorithm for the resource constrained shortest path problem. *Networks*, 19, 379-394.
- Beasley, J., Christofides, N. (1997). Vehicle routing with a sparse feasibility graph. *European Journal of Operational Research*, 98, 499-511.

- Beltrami, E., Bodin, L. (1974). Networks and Vehicle Routing for Municipal Waste Collection, *Networks*, 4(1), 65-94.
- Bertazzi, L., Palletta, G., Speranza M.G. (2005). Minimizing the total cost in an integrated vendor-managed inventory system. *Journal of Heuristics*, 11, 393-419.
- Bertazzi, L., Speranza, M.G., Savelsbergh, M.W.P. (2008). Inventory Routing. In: Golden B., Raghavan, S., Wasil, E. (Eds.). *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer.
- Bianco, L., Mingozzi, A., Ricciardelli, S. (1994). A set partitioning approach to the multiple depot vehicle scheduling problem. *Optimization Methods and Software*, 3(1), 163-194.
- Boland, N., Dethridge, J., Dumitrescu, I. (2006). Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Operations Research Letters*, 34, 58–68.
- Bostel, N., Dejax, P., Guez, P., Tricoire, F. (2008). Multiperiod Planning and Routing on a Rolling Horizon for Field Force Optimization Logistics. In: Golden B., Raghavan, S., Wasil, E. (Eds.). *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer, 503-525.
- Bradley, S.P., Hax, A.C., Magnanti, T.L. (1977). *Applied Mathematical Programming*. Addison-Wesley Publishing Company, Reading, MA.
- Campbell, A.M., Clarke, L., Kleywegt, A., Savelsbergh, M.W.P. (1998). Inventory routing. In: Crainic, T., Laporte, G. (Eds.). *Fleet Management and Logistics*, Kluwer Academic Publishers, Boston, 95-112.
- Campbell, A.M., Savelsbergh, M.W.P. (2004). A decomposition approach for the Inventory-Routing Problem. *Transportation Science*, 38(4), 488-502.
- Carpaneto, G., Amico, M.D., Fischetti, M., Toth, P. (1989). A Branch and Bound Algorithm for the Multiple Vehicle Scheduling Problem. *Networks*, 19, 531–548.
- Chabrier, A. (2006). Vehicle Routing Problem with elementary shortest path based column generation. *Computers & Operations Research*, 33, 2972–2990.
- Chabrier, A., Danna, E., and Le Pape, C. (2002). Cooperation entre generation de colonnes avec tournées sans cycle et recherche locale appliquee au routage de vehicules. In:

- Huitiemes Journees Nationales sur la Resolution de Problemes NP-Complets (JNPC 2002).*
- Chao, I.-M., Golden, B.L., Wasil, E.A. (1995). An improved heuristic for the period vehicle routing problem. *Networks*, 26 (1), 25–44.
- Christofides, N., (1979). The traveling salesman problem. In: Christofides, N., Mingozzi, R., Toth, P., Sandi, C., (Eds.). *Combinatorial Optimization*, Wiley, New York, 131-149.
- Christofides, N., Beasley, J. (1984). The period routing problem. *Networks*, 14, 237-256.
- Clarke, G., Wright, J., (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4), 568-581.
- Cook, T., Russell, R.A. (1978). A simulation and statistical analysis of stochastic vehicle routing with timing constraints. *Decision Science*, 9, 673–687.
- Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F. (2002). The VRP with Time Windows. In: Toth, P., Vigo, D., (Eds.), *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 157-193.
- Cordeau, J.F., Gendreau, M., Laporte, G. (1997). A Tabu Search Heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30, 105-119.
- Cordeau, J.F., Laporte, G., Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problem with time windows, *Journal of the Operational Research Society*, 57, 928-936.
- Cordone, R., Calvo, R.W. (1996). Note on time window constraints in routing problems. *Internal report 96.005*, Politecnico di Milano, Dipartimento di Electronica e Informazione, Milan, Italy.
- Cordone, R., Calvo, R.W. (1997). A heuristic for vehicle routing problem with time windows. *Internal Report 97.012*, Politecnico di Milano, Dipartimento di Electronica e Informazione, Milan, Italy.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. (2001). *Introduction to Algorithms*. New York, The MIT Press.
- Cornuejols, G., Nemhauser, G.L. (1978), “Tight bounds for Christofides traveling salesman heuristic”, *Mathematical Programming*, 14(1), 116-121.

- Cyrus, J.P. (1988). The Vehicle Scheduling Problem: Models, Complexity and Algorithms. Ph.D. Thesis, Technical University of Nova Scotia.
- Daganzo, C.F., Hall, C.F. (1993). A routing model for pickups and deliveries: No capacity restrictions on the secondary items. *Transportation Science*, 27, 315–329.
- 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.B, Wolfe, P. (1960), Decomposition principle for linear programs. *Operations research*, 8 (1), 101-111.
- Dantzig, G.B., Ramser, J.H. (1959). The truck dispatching problem. *Management Science*, 6(1), 80-91.
- Desaulniers, G., Desrosiers, J., Solomon, M.M. (2005). “*Column Generation*”, Springer Science and Business Media, Inc.
- Desaulniers, G., Lessard, F., Hadjar, A. (2006). Tabu search, generalized k-path inequalities, and partial elementarity for the vehicle routing problem with time windows. *Technical Report G-2006-45*, GERAD, Canada.
- Desrochers, M. (1988). An algorithm for the shortest path problem with resource constraints. *Technical Report G-88-27*, GERAD, Canada.
- Desrochers, M., Desrosiers, J., Solomon, M.M., (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40, 342–354.
- Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F. (1995). Time constrained routing and scheduling. In: Ball, M.O., Magnanti, T.L., Monma, C.L., Nemhauser, G.L. (Eds.). *Network Routing*, Handbooks in Operations Research and Management Science, 8, North-Holland, Amsterdam, The Netherlands, 35–139.
- Desrosiers, J., Lubbecke M.E. (2005). Selected Topics in Column Generation. *Operations Research*, 53(6), 1007–1023.
- Desrosiers, J., Soumis, F., Desrochers, M. (1984). Routing with time windows by column generation. *Networks*, 14, 545-565.
- Dijkstra, E.W. (1959). A note on two problems in Connexion with Graphs. *Numerische Mathematik*, 1, 269-271.

- Dror, M., (1994). Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research*, 42, 977-979.
- Dror, M., Ball, M., Golden, B. (1985). Computational comparison of algorithms for the inventory routing problem. *Annals of Operations Research*, 4, 3-23.
- Drummond, L.M.A., Ochi, L.S., Vianna, D.S. (2001). An asynchronous parallel metaheuristic for the period vehicle routing problem. *Future Generation Computer Systems*, 17(4), 379–386.
- Dumas, Y., Desrosiers, J. (1986). A shortest path problem for vehicle routing with pick-up, delivery, and time windows. *Technical Report G-86-09*, GERAD, Canada.
- Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks* 44, 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 Research on Transportation, Montreal.
- Fisher, M.L. (1985). An Application Oriented Guide to Lagrangean Relaxation. *Interfaces*, 15, 10-21.
- Ford, L.R., Fulkerson, D. R. (1958). A Suggested Computation For Maximal Multi-Commodity Network Flows, *Management Science*, 5 (1), 1958, 97-101
- Francis, P., Smilowitz, K. (2006). Modeling techniques for periodic vehicle routing problems. *Transportation Research Part B*, 40 (10), 872–884.
- Francis, P., Smilowitz, K., Tzur, M. (2006). The Period Vehicle Routing Problem with Service Choice, *Transportation Science*, 40, 439-454.
- Francis, P., Smilowitz, K., Tzur, M. (2008). The Period Vehicle Routing Problem and its Extensions. In: Golden, B., Raghavan, S., Wasil, E. (Eds.). *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer, 73-102.
- Garey, M.R., Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co, New York.
- Gaudioso, M., Paletta, G. (1992). A heuristic for the period vehicle routing problem. *Transportation Science*, 26 (2), 86–92.

- Gendreau, M., Laporte, G., Hertz, A. (1997). An Approximation Algorithm for the Traveling Salesman Problem with Backhauls. *Operation Research*, 45(4), 639-641.
- Gendreau, M., Potvin, J.-Y., Bräysy, O., Hasle, G., Løkketangen, A. (2008). Metaheuristics for the Vehicle Routing Problem and Its Extensions: A Categorized Bibliography, In: Golden, B., Raghavan, S., Wasil, E., (Eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges*, 49-72, Springer
- Geofrion, A.M. (1974). Lagrangean Relaxation and Its Uses in Linear Programming. *Mathematical Programming*, 2, 329-344.
- Gilmore, P.C., Gomory, R.E. (1961). A Linear Programming Approach to the Cutting Stock Problem. *Operations Research*, 9(6), 849-859.
- Gilmore, P.C., Gomory, R.E. (1963). A Linear Programming Approach to the Cutting Stock Problem-Part II. *Operations Research*, 11(6), 863-888.
- 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.
- Gunnarsson, H., Rönnqvist, M., (2008). Solving a multi-period supply chain problem for a pulp company using heuristics - An application to Södra Cell AB. *International Journal of Production Economics*, 116(1), 75-94.
- Hadjiconstantinou, E, Baldacci, R. (1998). A multi-depot period vehicle routing problem arising in the utilities sector. *The Journal of the Operational Research Society*, 49, 1239-1248.
- Halse, K. (1992). Modeling and solving complex vehicle routing problems. *Ph.D Thesis*, Technical University of Denmark.
- Harvey, W.D., Ginsberg, M.L. (1995). Limited Discrepancy Search. In: Proceedings IJ-CAI 1995.
- Homberger, J., Gehring, H. (1999). Extended Solomon's VRPTW instances. Available from: [www.fernuni-hagen.de/WINF/touren/inhalte/probinst.htm](http://www.fernuni-hagen.de/WINF/touren/inhalte/probinst.htm).

- Homberger, J., Gehring, H. (2005). A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operational Research*, 162, 220-238.
- Houck, D., Picard, J., Queyranne, M., and Vemuganti, R. (1980). The travelling salesman problem as a constrained shortest path problem: Theory and computational experience. *Opsearch*, 17, 93-109.
- Huisman, D., Jans, R., Peeters, M. Wagelmans A., (2005). Combining Column Generation and Lagrangian Relaxation. In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.). *Column Generation*, Springer.
- Ichoua, S., Gendreau, M., Potvin, J.Y. (2006). Exploiting Knowledge about Future Demands for Real-Time Vehicle Dispatching. *Transportation Science*, 40(2), 211-225.
- Irnich, S. (2001). The shortest path problem with k-cycle elimination ( $k \geq 3$ ): Improving a branch and price algorithm for the VRPTW. In: *TRISTAN IV*, Volume 3, 571–574.
- Irnich, S., Villeneuve, D. (2003). The shortest path problem with resource constraints and k-cycle elimination for  $k > 3$ . *Technical Report G-2003-55*, GERAD, Canada.
- Jaillet, P., Bard, J., Huang, L., Dror, M. (2002). Delivery Cost Approximations for Inventory Routing Problems in a Rolling Horizon Framework. *Transportation Science*, 36(3), 292-300.
- Jepsen, M., Petersen, B., Spoorendonk, S., Pisinger., D. (2008). Subset-Row Inequalities Applied to the Vehicle-Routing Problem with Time Windows. *Operations Research*, 56 (2), 497-511.
- Kallehauge, B., Larsen, J., Madsen, B.G., Solomon M.M. (2005). Vehicle Routing Problem with Time Windows, In: Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.). *Column Generation*, Springer Science and Business Media, Inc.
- Kindervater, G., Savelsbergh, M.W.P. (1997). Vehicle routing: Handling edge exchanges. In: Lenstra, J., Aarts, E. (Eds.). *Local Search in Combinatorial Optimization*, Wiley, Chichester, 337–360.
- Kohl, N. (1995). Exact Methods for the Time Constrained Routing and Related Scheduling Problems. *PhD thesis (IMM-PHD-1995-16)*, Department of Mathematical Modeling, Technical University of Denmark.

- 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), 395-406.
- Kolen, A., Rinnooy-Kan, A., Trienekens, H. (1987). Vehicle routing with time windows. *Operations Research*, 35(2), 266-274.
- Kontoravdis, G., Bard, J.F. (1995). A GRASP for the vehicle routing problem with time windows. *ORSA Journal on Computing*, 7, 10–23.
- Laporte, G., Osman, I.H. (1996). Routing problems: A bibliography. *Annals of Operation Research*, 61(1), 227-262.
- Larsen, J. (2001). Parallelization of the Vehicle Routing Problem with Time Windows. *PhD thesis (IMM-PHD-2001-62)*, Department of Mathematical Modeling, Technical University of Denmark.
- Lau, H. C., Liu, Q., Ono, H. (2002). Integrating local search and network flow to solve the inventory routing problem. In: 18th National Conference on Artificial intelligence, American Association for Artificial Intelligence, 9-14.
- Lau, H.C., Lim, A., Liu, Q.Z. (2000). Solving a Supply Chain Optimization Problem Collaboratively. In: 17th National Conference on Artificial Intelligence, American Association for Artificial Intelligence, 780–785.
- Lawler, E. L., Wood, D. E. (1966). Branch-and-bound methods: A survey. *Operations research*, 14 (4), 699–719.
- Lee, E.K., Mitchell, J.E. (2001). Integer Programming: Branch-and-Bound Methods. In: *Encyclopedia of Optimization*, Volume II, Kluwer Academic Publishers, 509-519.
- Lenstra, J.K., Rinnooy Kan, A.H.G. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11, 221-227.
- Luenberger, D.G. (1989). *Linear and Nonlinear Programming*. Addison-Wesley, Reading, MA.
- Mladenovic, N., Hansen, P. (1997). Variable neighbourhood search. *Computers and Operations Research*, 24, 1097–1100.

- Mourgaya, M., Vanderbeck, F. (2007). Column generation based heuristic for tactical planning in multi-period vehicle routing. *European Journal of Operational Research*, 183(3), 1028-1041.
- Newman, A.M., Yano, C.A., Kaminsky, P.M. (2005). Third Party Logistics planning with routing and inventory costs. In: Geunes, J., Pardalos, P.M. (Eds.). *Supply Chain Optimization*, Kluwer Academic Publishers, 87-122.
- Ninikas, G., Minis, I. (2011). A column generation technique for the Vehicle Routing Problem with Dynamic Pickups (VRPDP), working paper.
- Nori, V. (1999). Algorithms for dynamic and stochastic logistics problems. *PhD Thesis*. Department of Industrial and Systems Engineering, Georgia Institute of Technology, USA.
- Papadimitriou, C.H., Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*, Dover.
- Pirkwieser S., Raidl, G.R. (2009). A column generation approach for the periodic vehicle routing problem with time windows. In: M.G. Scutellà *et al.* (Eds.), Proceedings of the International Network Optimization Conference 2009, Pisa, Italy, April, 26-29.
- 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, 331–340.
- Ravichandran, N. (2007). A finite horizon inventory model: An operational framework. *International Journal of Production Economics*, 108, 406-415.
- Righini, G., Salani, M. (2005). New dynamic programming algorithms for the resource-constrained elementary shortest path problem. *Technical Report 66*, University of Milano, Italy.
- Russell, R.A. (1977). An effective heuristic for the m-tour traveling salesman problem with some side conditions. *Operations Research*, 25, 517–524.
- Russell, R.A. (1995). Hybrid heuristics for the vehicle routing problem with time windows. *Transportation Science*, 29, 156–166.
- Russell, R.A., Gribbin, D. (1991). A multiphase approach to the period routing problem. *Networks*, 21(7), 747–765.

- Russell, R.A., Igo, W. (1979). An assignment routing problem. *Networks*, 9(1), 1–17.
- Savelsbergh, M.W.P. (1985). Local search in routing problems with time windows. *Annals of Operations Research*, 4, 285–305.
- Savelsbergh, M.W.P. (1990). An efficient implementation of local search algorithms for constrained routing problems. *European Journal of Operational Research*, 47, 75–85.
- Savelsbergh, M.W.P. (1992). The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing*, 4, 146–154.
- 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, 161–174.
- Solomon, M.M. (1987). Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, 35, 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.
- Soumis, F. (1997). Decomposition and column generation. In: Amico, M.D., Maffioli, F., Martello, S. (Eds.). *Annotated Bibliographies in Combinatorial Optimization*, John Wiley and Sons, Chichester, 115–126.
- Taillard, E.D. (1999). A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO*, 33, 1-14.
- Tan, C.C.R., Beasley, J. (1984). A heuristic algorithm for the period vehicle routing problem, *Omega*, 12(5), 497-504.
- Teng, S.Y., Ong, H.L., Huang, H.C. (2006). Heuristic algorithms for visiting the customers in a rolling schedule environment. *OR Spectrum*, 28(2), 241-266.
- Thompson, P.M., Psaraftis, H.N. (1993). Cyclic transfer algorithms for multi-vehicle routing and scheduling problems. *Operations Research*, 41, 935–946.
- Toth, P., Vigo, D. (2002). *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, USA.

- Tricoire, F. (2006). Optimisation Des Tournées De Véhicules Et De Personnels De Maintenance: Application A La Distribution Et Au Traitement Des Eaux, *PhD Thesis*, Université de Nantes.
- Tricoire, F. (2007). Vehicle and personnel routing optimization in the service sector: application to water distribution and treatment, *4OR*, 5, 165-168.
- Wen, M., Cordeau, J-F, Laporte, G., Larsen, J. (2009). The dynamic multi-period vehicle routing problem, *Computers & Operations Research*, In Press.
- Zäpfel, G., Bögl, M. (2008). Multi-period vehicle routing and crew scheduling with outsourcing options, *International Journal of Production Economics*, 113(2), 980-996.
- Ziegelmann, M. (2001). Constrained Shortest Paths and Related Problems. *PhD Thesis*, Universität des Saarlandes, 2001.



## APPENDICES

## APPENDIX A: LOWER BOUNDS AND COMPUTATIONAL TIMES FOR ALL CG TECHNIQUES

Table A.1 presents the results for the instances solved in Chapter 4. Specifically, Table A.1 provides the problem ID, the period window pattern (1 to 9), the maximum allowable number of vehicles per period and: (a) the lower bound obtained and (b) the needed computational time for each one of the tested CG methods, i.e. the FULL, CLONE, UNIFIED and Parallel methods. Note that lower bound values highlighted in grey are different to the values provided by the other CG methods due to the presence of the LDS procedure within the column generation procedure.

Table A.1: Lower Bounds and Computational Time

Probl.	Pat.	Vehicle Limit	Lower Bound				Computational Time (sec)			
			FULL	CLON	UNI	PARA	FULL	CLON	UNIF	PARA
r101	1	6	1541,30	1541,30	1541,30	1541,30	<b>2,28</b>	<b>0,70</b>	<b>2,59</b>	<b>1,14</b>
r101	2	6	1253,90	1253,90	1253,90	1253,90	<b>3,95</b>	<b>3,12</b>	<b>2,59</b>	<b>2,27</b>
r101	3	4*	1191,64	1191,64	1191,64	1191,64	<b>4,99</b>	<b>3,45</b>	<b>4,11</b>	<b>3,33</b>
r101	4	4*	1144,33	1144,33	1144,33	1144,33	<b>6,91</b>	<b>6,27</b>	<b>6,73</b>	<b>4,24</b>
r101	5	4*	1119,70	1119,70	1119,70	1119,70	<b>7,43</b>	<b>5,83</b>	<b>5,41</b>	<b>5,89</b>
r101	6	4*	1048,37	1048,37	1048,37	1048,37	<b>9,53</b>	<b>6,00</b>	<b>7,00</b>	<b>6,46</b>
r101	7	4*	1048,37	1048,37	1048,37	1048,37	<b>12,18</b>	<b>10,48</b>	<b>6,09</b>	<b>6,66</b>
r101	8	4*	1048,37	1048,37	1048,37	1048,37	<b>11,89</b>	<b>7,24</b>	<b>7,27</b>	<b>6,67</b>
r101	9	4*	1048,37	1048,37	1048,37	1048,37	<b>19,05</b>	<b>6,91</b>	<b>6,96</b>	<b>8,24</b>
r102	1	4	1410,60	1410,60	1414,80	1410,60	<b>2,00</b>	<b>2,83</b>	<b>3,57</b>	<b>2,69</b>
r102	2	3	1168,30	1168,30	1168,30	1168,30	<b>5,78</b>	<b>4,15</b>	<b>6,90</b>	<b>4,17</b>
r102	3	3	1104,04	1104,04	1103,53	1104,04	<b>8,92</b>	<b>9,33</b>	<b>11,96</b>	<b>6,84</b>
r102	4	3	1050,90	1050,90	1050,60	1050,90	<b>23,08</b>	<b>18,88</b>	<b>19,68</b>	<b>15,29</b>
r102	5	3	1034,76	1034,76	1033,54	1034,76	<b>30,22</b>	<b>23,63</b>	<b>29,79</b>	<b>24,86</b>
r102	6	3	923,15	923,15	923,15	923,15	<b>17,98</b>	<b>12,42</b>	<b>14,45</b>	<b>12,79</b>
r102	7	3	914,00	914,00	914,00	914,00	<b>22,32</b>	<b>15,83</b>	<b>15,57</b>	<b>14,07</b>
r102	8	3	914,00	914,00	914,00	914,00	<b>28,17</b>	<b>15,49</b>	<b>16,72</b>	<b>18,13</b>
r102	9	3	914,00	914,00	914,00	914,00	<b>30,00</b>	<b>17,19</b>	<b>15,43</b>	<b>20,10</b>
r103	1	4	1277,40	1277,40	1283,00	1277,40	<b>1,64</b>	<b>1,78</b>	<b>4,69</b>	<b>1,44</b>
r103	2	3	1020,40	1020,40	1021,30	1020,40	<b>6,42</b>	<b>6,94</b>	<b>14,39</b>	<b>6,28</b>
r103	3	3	965,10	965,10	965,10	965,10	<b>18,53</b>	<b>15,99</b>	<b>26,89</b>	<b>15,12</b>
r103	4	3	870,67	870,67	870,67	870,67	<b>55,15</b>	<b>33,71</b>	<b>64,44</b>	<b>40,38</b>
r103	5	3	858,20	858,20	857,98	858,20	<b>79,19</b>	<b>86,17</b>	<b>129,04</b>	<b>72,67</b>
r103	6	3	785,97	785,97	785,97	785,97	<b>64,90</b>	<b>53,04</b>	<b>64,87</b>	<b>40,02</b>
r103	7	3	774,23	774,23	774,23	774,23	<b>50,61</b>	<b>40,76</b>	<b>31,74</b>	<b>26,50</b>
r103	8	3	774,23	774,23	774,23	774,23	<b>62,15</b>	<b>37,33</b>	<b>39,00</b>	<b>32,71</b>
r103	9	3	774,23	774,23	774,23	774,23	<b>74,81</b>	<b>33,85</b>	<b>28,09</b>	<b>33,89</b>
r104	1	2	1115,90	1115,90	1126,85	1115,90	<b>3,35</b>	<b>3,25</b>	<b>9,16</b>	<b>2,57</b>
r104	2	2	831,70	831,70	831,70	831,70	<b>31,39</b>	<b>37,61</b>	<b>49,85</b>	<b>24,02</b>
r104	3	2	788,91	788,91	788,91	788,91	<b>118,05</b>	<b>111,14</b>	<b>169,78</b>	<b>91,98</b>
r104	4	2	N/A							

Probl.	Pat.	Vehicle Limit	Lower Bound		Computational Time (sec)					
			FULL	CLON	UNI	PARA	FULL	CLON	UNIF	PARA
r104	5	2	N/A							
r104	6	2	630,43	630,43	630,43	630,43	1747,99	908,94	838,65	1273,22
r104	7	2	624,08	624,08	624,08	624,08	1366,01	880,69	649,55	842,42
r104	8	2	624,08	624,08	624,08	624,08	1440,61	841,43	800,14	974,95
r104	9	2	624,08	624,08	624,08	624,08	2020,35	761,73	567,48	1243,58
r105	1	3	1465,20	1465,20	1465,20	1465,20	0,85	0,79	1,85	1,40
r105	2	3	1157,00	1157,00	1164,50	1157,00	2,57	2,34	4,84	2,30
r105	3	3	1080,40	1080,40	1077,60	1080,40	6,25	4,88	8,07	4,73
r105	4	3	1024,20	1024,20	1018,66	1024,20	11,16	10,60	14,34	8,84
r105	5	3	1004,59	1004,59	992,60	1004,59	12,71	11,50	12,20	11,10
r105	6	3	901,66	901,66	901,66	901,66	12,02	10,28	8,88	8,61
r105	7	3	897,12	897,12	897,12	897,12	18,20	7,15	7,22	9,90
r105	8	3	897,12	897,12	897,12	897,12	20,29	12,74	9,21	10,46
r105	9	3	897,12	897,12	897,12	897,12	20,79	9,77	9,25	10,69
r106	1	4	1320,40	1320,40	1323,60	1320,40	1,50	1,45	3,44	1,56
r106	2	3	1041,95	1041,95	1041,95	1041,95	5,71	5,05	12,81	4,34
r106	3	3	958,19	958,19	958,19	958,19	10,54	8,63	11,87	7,73
r106	4	3	876,93	876,93	876,93	876,93	21,33	18,97	33,99	16,14
r106	5	3	850,50	850,50	849,34	850,50	33,00	31,15	50,53	27,05
r106	6	3	796,88	796,88	796,88	796,88	39,72	29,05	24,58	23,62
r106	7	3	796,37	796,37	796,37	796,37	43,61	30,10	23,89	21,86
r106	8	3	796,37	796,37	796,37	796,37	50,72	24,00	31,03	23,25
r106	9	3	796,37	796,37	796,37	796,37	59,57	24,99	23,80	26,56
r107	1	4	1221,00	1221,00	1225,20	1221,00	3,25	2,01	3,89	2,07
r107	2	3	946,23	946,23	946,66	946,23	10,67	11,09	22,76	7,84
r107	3	3	873,80	873,80	873,80	873,80	21,18	24,80	46,28	15,43
r107	4	3	792,26	792,26	792,26	792,26	43,51	45,37	119,66	32,58
r107	5	3	762,59	762,59	761,84	762,59	86,65	78,33	232,47	74,69
r107	6	3	714,41	714,41	714,41	714,41	83,64	64,33	72,96	46,87
r107	7	3	712,26	712,26	712,26	712,26	113,00	65,33	69,93	53,86
r107	8	3	712,26	712,26	712,26	712,26	172,01	78,49	73,78	74,21
r107	9	3	712,26	712,26	712,26	712,26	166,78	74,94	62,20	75,67
r108	1	3	1113,70	1113,70	1123,65	1113,70	4,23	3,88	7,02	2,85
r108	2	3	793,40	793,40	793,40	793,40	17,03	20,63	28,35	11,22
r108	3	3	745,66	745,66	745,66	745,66	44,22	48,32	99,60	27,88
r108	4	3	657,40	657,40	657,40	657,40	193,54	197,47	543,23	131,87
r108	5	3	N/A							
r108	6	3	599,70	599,70	599,70	599,70	2393,63	1846,30	2375,52	1737,70
r108	7	3	599,70	599,70	599,70	599,70	3443,39	1429,92	1571,16	2261,83
r108	8	3	599,70	599,70	599,70	599,70	4860,45	1709,37	1596,94	3318,27
r108	9	3	599,70	599,70	599,70	599,70	4755,93	1934,00	1309,59	3930,01
r109	1	3	1286,00	1286,00	1286,00	1286,00	1,40	1,39	2,97	2,20
r109	2	3	1001,93	1001,93	1001,93	1001,93	5,33	5,28	6,79	37,46
r109	3	3	908,95	908,95	908,95	908,95	11,60	9,37	11,67	18,89
r109	4	3	829,89	829,89	829,87	829,89	21,83	15,85	20,54	33,42
r109	5	3	812,86	812,86	812,86	812,86	20,28	18,10	17,69	31,68
r109	6	3	781,27	781,27	781,27	781,27	21,94	14,28	14,97	20,85
r109	7	3	780,34	780,34	780,34	780,34	32,51	21,43	18,59	26,06
r109	8	3	780,34	780,34	780,34	780,34	43,22	19,55	16,52	29,12
r109	9	3	780,34	780,34	780,34	780,34	43,52	21,58	20,69	24,38
r110	1	3*	1237,77	1237,77	1237,77	1237,77	1,64	1,78	2,60	1,59
r110	2	3*	927,68	927,68	927,68	927,68	9,61	8,27	10,80	7,43
r110	3	2*	899,55	899,55	899,55	899,55	36,65	30,71	52,43	26,94
r110	4	2*	850,48	850,48	850,48	850,48	61,93	57,17	135,11	41,80

Probl.	Pat.	Vehicle Limit	Lower Bound				Computational Time (sec)			
			FULL	CLON	UNI	PARA	FULL	CLON	UNIF	PARA
r110	5	2*	846,53	846,53	846,53	846,53	<b>77,43</b>	<b>80,26</b>	<b>181,05</b>	<b>59,67</b>
r110	6	2	709,63	709,63	709,63	709,63	<b>95,81</b>	<b>58,71</b>	<b>69,15</b>	<b>53,46</b>
r110	7	2	700,06	700,06	700,06	700,06	<b>65,36</b>	<b>42,40</b>	<b>40,50</b>	<b>35,55</b>
r110	8	2	700,06	700,06	700,06	700,06	<b>91,98</b>	<b>46,07</b>	<b>43,37</b>	<b>46,76</b>
r110	9	2	700,06	700,06	700,06	700,06	<b>77,86</b>	<b>38,95</b>	<b>32,86</b>	<b>34,99</b>
r111	1	3	1212,75	1212,75	1226,70	1212,75	<b>1,88</b>	<b>1,96</b>	<b>4,35</b>	<b>1,71</b>
r111	2	3	915,66	915,66	915,66	915,66	<b>9,97</b>	<b>9,81</b>	<b>15,69</b>	<b>7,22</b>
r111	3	3	851,60	851,60	851,60	851,60	<b>22,66</b>	<b>17,84</b>	<b>31,55</b>	<b>15,61</b>
r111	4	3	759,41	759,41	759,29	759,41	<b>47,38</b>	<b>32,17</b>	<b>88,30</b>	<b>36,56</b>
r111	5	3	736,29	736,29	736,28	736,29	<b>118,25</b>	<b>101,80</b>	<b>160,54</b>	<b>106,94</b>
r111	6	3	701,29	701,29	701,29	701,29	<b>90,44</b>	<b>51,41</b>	<b>53,66</b>	<b>50,80</b>
r111	7	3	701,29	701,29	701,29	701,29	<b>100,83</b>	<b>60,04</b>	<b>58,03</b>	<b>48,13</b>
r111	8	3	701,29	701,29	701,29	701,29	<b>137,38</b>	<b>80,27</b>	<b>61,13</b>	<b>62,00</b>
r111	9	3	701,29	701,29	701,29	701,29	<b>165,64</b>	<b>78,57</b>	<b>62,08</b>	<b>75,68</b>
r112	1	3	1112,90	1112,90	1112,90	1112,90	<b>2,99</b>	<b>2,86</b>	<b>8,51</b>	<b>2,72</b>
r112	2	3	815,23	815,23	815,23	815,23	<b>13,82</b>	<b>12,52</b>	<b>35,91</b>	<b>9,74</b>
r112	3	3	749,17	749,17	749,17	749,17	<b>33,95</b>	<b>35,61</b>	<b>76,95</b>	<b>21,41</b>
r112	4	3	669,55	669,55	669,55	669,55	<b>85,06</b>	<b>99,81</b>	<b>117,77</b>	<b>52,69</b>
r112	5	3	644,35	644,35	644,35	644,35	<b>332,13</b>	<b>401,95</b>	<b>561,26</b>	<b>330,84</b>
r112	6	3	619,85	619,85	619,85	619,85	<b>337,61</b>	<b>202,77</b>	<b>237,22</b>	<b>199,95</b>
r112	7	3	619,85	619,85	619,85	619,85	<b>346,97</b>	<b>243,34</b>	<b>289,61</b>	<b>192,90</b>
r112	8	3	619,85	619,85	619,85	619,85	<b>537,72</b>	<b>283,26</b>	<b>250,78</b>	<b>284,49</b>
r112	9	3	619,85	619,85	619,85	619,85	<b>647,97</b>	<b>256,85</b>	<b>193,63</b>	<b>338,56</b>
c101	1	2	455,30	455,30	455,30	455,30	<b>3,42</b>	<b>2,25</b>	<b>3,66</b>	<b>2,78</b>
c101	2	2	367,40	367,40	367,40	367,40	<b>8,24</b>	<b>7,42</b>	<b>8,17</b>	<b>5,79</b>
c101	3	2	367,40	367,40	367,40	367,40	<b>16,04</b>	<b>9,46</b>	<b>12,67</b>	<b>10,02</b>
c101	4	2	367,40	367,40	367,40	367,40	<b>11,47</b>	<b>11,46</b>	<b>12,34</b>	<b>8,36</b>
c101	5	2	367,40	367,40	367,40	367,40	<b>30,73</b>	<b>13,87</b>	<b>16,66</b>	<b>21,09</b>
c101	6	2	367,40	367,40	367,40	367,40	<b>26,85</b>	<b>14,31</b>	<b>13,83</b>	<b>16,80</b>
c101	7	2	367,40	367,40	367,40	367,40	<b>28,42</b>	<b>15,46</b>	<b>19,52</b>	<b>15,95</b>
c101	8	2	367,40	367,40	367,40	367,40	<b>54,20</b>	<b>19,85</b>	<b>22,28</b>	<b>30,61</b>
c101	9	2	367,40	367,40	367,40	367,40	<b>38,54</b>	<b>18,58</b>	<b>16,57</b>	<b>17,92</b>
c102	1	3	454,30	454,30	454,30	454,30	<b>3,43</b>	<b>3,15</b>	<b>8,49</b>	<b>3,20</b>
c102	2	3	366,40	366,40	366,40	366,40	<b>23,75</b>	<b>15,76</b>	<b>25,37</b>	<b>18,04</b>
c102	3	3	366,40	366,40	366,40	366,40	<b>27,32</b>	<b>22,05</b>	<b>48,60</b>	<b>17,60</b>
c102	4	3	366,40	366,40	366,40	366,40	<b>36,42</b>	<b>24,01</b>	<b>44,56</b>	<b>25,78</b>
c102	5	3	366,40	366,40	366,40	366,40	<b>44,16</b>	<b>33,38</b>	<b>33,12</b>	<b>29,44</b>
c102	6	3	366,40	366,40	366,40	366,40	<b>93,12</b>	<b>36,37</b>	<b>43,96</b>	<b>53,79</b>
c102	7	3	366,40	366,40	366,40	366,40	<b>72,47</b>	<b>43,69</b>	<b>45,87</b>	<b>37,66</b>
c102	8	3	366,40	366,40	366,40	366,40	<b>74,22</b>	<b>40,30</b>	<b>41,83</b>	<b>38,23</b>
c102	9	3	366,40	366,40	366,40	366,40	<b>151,13</b>	<b>63,49</b>	<b>68,82</b>	<b>71,00</b>
c103	1	2	446,42	446,42	446,42	446,42	<b>4,98</b>	<b>5,01</b>	<b>14,32</b>	<b>4,16</b>
c103	2	2	366,40	366,40	366,40	366,40	<b>123,50</b>	<b>49,66</b>	<b>167,96</b>	<b>80,06</b>
c103	3	2	366,40	366,40	366,40	366,40	<b>121,13</b>	<b>135,61</b>	<b>312,35</b>	<b>76,02</b>
c103	4	2	366,40	366,40	366,40	366,40	<b>159,94</b>	<b>185,25</b>	<b>185,76</b>	<b>102,12</b>
c103	5	2	366,40	366,40	366,40	366,40	<b>144,26</b>	<b>155,49</b>	<b>372,38</b>	<b>93,79</b>
c103	6	2	366,40	366,40	366,40	366,40	<b>245,08</b>	<b>104,49</b>	<b>239,43</b>	<b>128,22</b>
c103	7	2	366,40	366,40	366,40	366,40	<b>254,65</b>	<b>175,96</b>	<b>537,15</b>	<b>128,74</b>
c103	8	2	366,40	366,40	366,40	366,40	<b>545,36</b>	<b>235,95</b>	<b>444,50</b>	<b>321,44</b>
c103	9	2	366,40	366,40	366,40	366,40	<b>904,25</b>	<b>141,25</b>	<b>137,84</b>	<b>472,39</b>
c104	1	2	422,69	422,69	422,69	422,69	<b>48,89</b>	<b>48,51</b>	<b>92,81</b>	<b>48,15</b>
c104	2	2	N/A							
c104	3	2	N/A							
c104	4	2	N/A							

Probl.	Pat.	Vehicle Limit	Lower Bound		Computational Time (sec)					
			FULL	CLON	UNI	PARA	FULL	CLON	UNIF	PARA
c104	5	2	N/A							
c104	6	2	N/A							
c104	7	2	N/A							
c104	8	2	N/A							
c104	9	2	N/A							
c105	1	2	455,30	455,30	455,30	455,30	<b>2,37</b>	<b>2,31</b>	<b>4,28</b>	<b>22,84</b>
c105	2	2	367,40	367,40	367,40	367,40	<b>8,91</b>	<b>8,74</b>	<b>14,54</b>	<b>26,80</b>
c105	3	2	367,40	367,40	367,40	367,40	<b>15,58</b>	<b>15,20</b>	<b>17,51</b>	<b>29,53</b>
c105	4	2	367,40	367,40	367,40	367,40	<b>17,94</b>	<b>14,27</b>	<b>13,80</b>	<b>24,66</b>
c105	5	2	367,40	367,40	367,40	367,40	<b>22,00</b>	<b>14,64</b>	<b>24,23</b>	<b>18,75</b>
c105	6	2	367,40	367,40	367,40	367,40	<b>24,93</b>	<b>18,43</b>	<b>18,58</b>	<b>15,58</b>
c105	7	2	367,40	367,40	367,40	367,40	<b>43,59</b>	<b>23,27</b>	<b>26,79</b>	<b>24,96</b>
c105	8	2	367,40	367,40	367,40	367,40	<b>49,28</b>	<b>18,40</b>	<b>22,57</b>	<b>26,13</b>
c105	9	2	367,40	367,40	367,40	367,40	<b>67,04</b>	<b>25,69</b>	<b>30,95</b>	<b>32,42</b>
c106	1	2	454,50	454,50	454,50	454,50	<b>2,50</b>	<b>3,16</b>	<b>3,55</b>	<b>2,81</b>
c106	2	2	367,40	367,40	367,40	367,40	<b>7,57</b>	<b>8,04</b>	<b>11,64</b>	<b>6,56</b>
c106	3	2	367,40	367,40	367,40	367,40	<b>16,86</b>	<b>11,59</b>	<b>13,85</b>	<b>11,85</b>
c106	4	2	367,40	367,40	367,40	367,40	<b>14,02</b>	<b>15,19</b>	<b>21,53</b>	<b>11,40</b>
c106	5	2	367,40	367,40	367,40	367,40	<b>20,76</b>	<b>11,12</b>	<b>19,50</b>	<b>15,43</b>
c106	6	2	367,40	367,40	367,40	367,40	<b>33,00</b>	<b>16,36</b>	<b>21,56</b>	<b>19,52</b>
c106	7	2	367,40	367,40	367,40	367,40	<b>35,94</b>	<b>18,50</b>	<b>20,85</b>	<b>20,64</b>
c106	8	2	367,40	367,40	367,40	367,40	<b>50,24</b>	<b>24,55</b>	<b>25,80</b>	<b>26,52</b>
c106	9	2	367,40	367,40	367,40	367,40	<b>54,21</b>	<b>18,85</b>	<b>21,00</b>	<b>27,46</b>
c107	1	2	455,30	455,30	455,30	455,30	<b>2,36</b>	<b>2,40</b>	<b>4,80</b>	<b>2,84</b>
c107	2	2	367,40	367,40	367,40	367,40	<b>11,53</b>	<b>10,18</b>	<b>9,90</b>	<b>10,24</b>
c107	3	2	367,40	367,40	367,40	367,40	<b>20,63</b>	<b>16,50</b>	<b>17,24</b>	<b>15,77</b>
c107	4	2	367,40	367,40	367,40	367,40	<b>22,51</b>	<b>16,92</b>	<b>23,51</b>	<b>16,17</b>
c107	5	2	367,40	367,40	367,40	367,40	<b>26,69</b>	<b>16,15</b>	<b>23,44</b>	<b>18,46</b>
c107	6	2	367,40	367,40	367,40	367,40	<b>29,27</b>	<b>22,19</b>	<b>28,30</b>	<b>17,85</b>
c107	7	2	367,40	367,40	367,40	367,40	<b>42,01</b>	<b>20,09</b>	<b>30,81</b>	<b>24,84</b>
c107	8	2	367,40	367,40	367,40	367,40	<b>47,50</b>	<b>23,30</b>	<b>30,50</b>	<b>25,70</b>
c107	9	2	367,40	367,40	367,40	367,40	<b>56,66</b>	<b>23,64</b>	<b>23,94</b>	<b>27,35</b>
c108	1	2	448,90	448,90	448,90	448,90	<b>3,24</b>	<b>3,46</b>	<b>8,37</b>	<b>3,85</b>
c108	2	2	367,40	367,40	367,40	367,40	<b>14,62</b>	<b>16,12</b>	<b>22,55</b>	<b>10,84</b>
c108	3	2	367,40	367,40	367,40	367,40	<b>32,00</b>	<b>23,74</b>	<b>41,30</b>	<b>20,98</b>
c108	4	2	367,40	367,40	367,40	367,40	<b>33,98</b>	<b>26,56</b>	<b>38,18</b>	<b>22,17</b>
c108	5	2	367,40	367,40	367,40	367,40	<b>44,53</b>	<b>25,70</b>	<b>51,14</b>	<b>30,88</b>
c108	6	2	367,40	367,40	367,40	367,40	<b>55,49</b>	<b>27,34</b>	<b>44,67</b>	<b>29,82</b>
c108	7	2	367,40	367,40	367,40	367,40	<b>61,36</b>	<b>61,12</b>	<b>52,48</b>	<b>31,64</b>
c108	8	2	367,40	367,40	367,40	367,40	<b>99,90</b>	<b>35,67</b>	<b>49,13</b>	<b>47,48</b>
c108	9	2	367,40	367,40	367,40	367,40	<b>90,87</b>	<b>38,17</b>	<b>44,77</b>	<b>43,97</b>
c109	1	2	422,98	422,98	422,98	422,98	<b>4,38</b>	<b>4,42</b>	<b>9,48</b>	<b>3,86</b>
c109	2	2	367,40	367,40	367,40	367,40	<b>41,32</b>	<b>39,73</b>	<b>64,21</b>	<b>29,58</b>
c109	3	2	367,40	367,40	367,40	367,40	<b>68,45</b>	<b>41,73</b>	<b>69,76</b>	<b>40,91</b>
c109	4	2	367,40	367,40	367,40	367,40	<b>58,94</b>	<b>52,00</b>	<b>95,52</b>	<b>36,62</b>
c109	5	2	367,40	367,40	367,40	367,40	<b>85,95</b>	<b>59,83</b>	<b>88,38</b>	<b>54,50</b>
c109	6	2	367,40	367,40	367,40	367,40	<b>120,98</b>	<b>53,67</b>	<b>96,59</b>	<b>61,34</b>
c109	7	2	367,40	367,40	367,40	367,40	<b>95,71</b>	<b>65,56</b>	<b>97,20</b>	<b>44,78</b>
c109	8	2	367,40	367,40	367,40	367,40	<b>112,47</b>	<b>71,03</b>	<b>108,87</b>	<b>47,37</b>
c109	9	2	367,40	367,40	367,40	367,40	<b>212,11</b>	<b>64,51</b>	<b>130,34</b>	<b>109,58</b>
rc101	1	4	1300,80	1300,80	1300,80	1300,80	<b>1,13</b>	<b>1,14</b>	<b>1,97</b>	<b>1,97</b>
rc101	2	3	1061,38	1061,38	1061,38	1061,38	<b>4,42</b>	<b>4,47</b>	<b>5,13</b>	<b>3,41</b>
rc101	3	3	995,29	995,29	995,29	995,29	<b>7,01</b>	<b>5,32</b>	<b>6,11</b>	<b>5,37</b>
rc101	4	2	N/A							

Probl.	Pat.	Vehicle Limit	Lower Bound				Computational Time (sec)			
			FULL	CLON	UNI	PARA	FULL	CLON	UNIF	PARA
rc101	5	2	903,54	903,54	903,54	903,54	<b>12,51</b>	<b>11,60</b>	<b>9,59</b>	<b>9,01</b>
rc101	6	2	855,02	855,02	855,02	855,02	<b>13,23</b>	<b>9,26</b>	<b>8,79</b>	<b>7,48</b>
rc101	7	2	855,02	855,02	855,02	855,02	<b>17,27</b>	<b>10,65</b>	<b>11,07</b>	<b>9,87</b>
rc101	8	2	855,02	855,02	855,02	855,02	<b>25,59</b>	<b>11,58</b>	<b>12,71</b>	<b>12,96</b>
rc101	9	2	855,02	855,02	855,02	855,02	<b>28,91</b>	<b>12,33</b>	<b>13,08</b>	<b>14,25</b>
rc102	1	3	1205,20	1205,20	1205,20	1205,20	<b>1,82</b>	<b>1,79</b>	<b>4,82</b>	<b>1,96</b>
rc102	2	2	951,57	951,58	951,58	951,57	<b>5,27</b>	<b>6,17</b>	<b>7,27</b>	<b>4,13</b>
rc102	3	2	895,47	895,47	895,48	895,47	<b>8,41</b>	<b>10,96</b>	<b>9,20</b>	<b>6,28</b>
rc102	4	2	847,15	847,15	847,15	847,15	<b>13,15</b>	<b>17,55</b>	<b>18,81</b>	<b>8,68</b>
rc102	5	2	726,81	726,81	726,82	726,81	<b>22,03</b>	<b>21,56</b>	<b>14,47</b>	<b>16,04</b>
rc102	6	2	726,82	726,82	726,82	726,82	<b>22,20</b>	<b>17,36</b>	<b>15,77</b>	<b>12,08</b>
rc102	7	2	726,81	726,81	726,81	726,81	<b>44,92</b>	<b>18,40</b>	<b>18,65</b>	<b>21,60</b>
rc102	8	2	726,82	726,81	726,81	726,82	<b>42,73</b>	<b>17,97</b>	<b>17,33</b>	<b>19,66</b>
rc102	9	2	726,82	726,82	726,81	726,82	<b>47,40</b>	<b>24,23</b>	<b>24,25</b>	<b>21,49</b>
rc103	1	3	1140,60	1140,60	1140,60	1140,60	<b>2,21</b>	<b>2,19</b>	<b>5,80</b>	<b>1,77</b>
rc103	2	2	869,32	869,33	869,32	869,32	<b>12,92</b>	<b>18,11</b>	<b>17,22</b>	<b>8,14</b>
rc103	3	2	829,58	829,58	829,58	829,58	<b>24,34</b>	<b>20,10</b>	<b>35,61</b>	<b>15,46</b>
rc103	4	2	750,38	750,38	750,38	750,38	<b>54,47</b>	<b>44,06</b>	<b>54,99</b>	<b>36,43</b>
rc103	5	2	650,28	650,28	650,28	650,28	<b>61,99</b>	<b>60,23</b>	<b>48,06</b>	<b>39,04</b>
rc103	6	2	650,28	650,28	650,28	650,28	<b>140,78</b>	<b>68,50</b>	<b>97,78</b>	<b>75,09</b>
rc103	7	2	650,28	650,28	650,28	650,28	<b>185,53</b>	<b>78,01</b>	<b>72,35</b>	<b>83,84</b>
rc103	8	2	650,28	650,28	650,28	650,28	<b>153,80</b>	<b>64,96</b>	<b>53,04</b>	<b>61,46</b>
rc103	9	2	650,28	650,28	650,28	650,28	<b>364,74</b>	<b>128,52</b>	<b>104,55</b>	<b>158,48</b>
rc104	1	3	1061,20	1061,20	1061,20	1061,20	<b>3,31</b>	<b>3,18</b>	<b>16,05</b>	<b>2,59</b>
rc104	2	2	778,10	778,10	778,10	778,10	<b>96,81</b>	<b>74,84</b>	<b>423,42</b>	<b>65,16</b>
rc104	3	2	721,80	721,80	721,80	721,80	<b>230,80</b>	<b>94,73</b>	<b>973,99</b>	<b>170,71</b>
rc104	4	2	649,60	649,60	649,60	649,60	<b>300,77</b>	<b>299,59</b>	<b>382,70</b>	<b>305,92</b>
rc104	5	2	550,80	550,80	550,80	550,80	<b>436,99</b>	<b>1014,36</b>	<b>342,81</b>	<b>348,71</b>
rc104	6	2	550,80	550,80	550,80	550,80	<b>680,42</b>	<b>990,32</b>	<b>606,68</b>	<b>413,14</b>
rc104	7	2	550,80	550,80	550,80	550,80	<b>4123,99</b>	<b>1614,31</b>	<b>579,05</b>	<b>2807,22</b>
rc104	8	2	550,80	550,80	550,80	550,80	<b>833,57</b>	<b>1229,37</b>	<b>447,24</b>	<b>469,30</b>
rc104	9	2	550,80	550,80	550,80	550,80	<b>2272,42</b>	<b>303,41</b>	<b>219,39</b>	<b>1331,80</b>
rc105	1	3	1243,80	1243,80	1243,80	1243,80	<b>1,87</b>	<b>2,09</b>	<b>2,99</b>	<b>2,03</b>
rc105	2	3	996,46	996,46	996,46	996,46	<b>7,82</b>	<b>8,16</b>	<b>7,24</b>	<b>5,01</b>
rc105	3	3	942,76	942,76	942,76	942,76	<b>11,04</b>	<b>8,92</b>	<b>12,59</b>	<b>6,88</b>
rc105	4	3	842,16	842,16	842,16	842,16	<b>13,50</b>	<b>15,43</b>	<b>10,33</b>	<b>9,06</b>
rc105	5	3	766,56	766,56	766,56	766,56	<b>21,41</b>	<b>12,16</b>	<b>20,00</b>	<b>15,23</b>
rc105	6	2	766,56	766,56	766,56	766,56	<b>29,90</b>	<b>15,83</b>	<b>18,53</b>	<b>15,75</b>
rc105	7	2	766,56	766,56	766,56	766,56	<b>37,42</b>	<b>18,31</b>	<b>16,06</b>	<b>18,31</b>
rc105	8	2	766,56	766,56	766,56	766,56	<b>40,10</b>	<b>19,51</b>	<b>18,01</b>	<b>18,97</b>
rc105	9	2	766,56	766,56	766,56	766,56	<b>44,23</b>	<b>20,52</b>	<b>18,24</b>	<b>19,53</b>
rc106	1	3	1133,80	1133,80	1133,80	1133,80	<b>2,31</b>	<b>2,04</b>	<b>2,69</b>	<b>2,36</b>
rc106	2	2	885,80	885,80	885,80	885,80	<b>8,24</b>	<b>7,20</b>	<b>10,11</b>	<b>4,87</b>
rc106	3	2	846,23	846,23	846,23	846,23	<b>17,70</b>	<b>8,71</b>	<b>10,61</b>	<b>9,18</b>
rc106	4	2	779,23	779,23	779,23	779,23	<b>24,20</b>	<b>17,07</b>	<b>14,21</b>	<b>12,90</b>
rc106	5	2	669,43	669,43	669,43	669,43	<b>28,12</b>	<b>18,71</b>	<b>14,87</b>	<b>16,20</b>
rc106	6	2	669,43	669,43	669,43	669,43	<b>36,15</b>	<b>21,46</b>	<b>21,12</b>	<b>18,32</b>
rc106	7	2	669,43	669,43	669,43	669,43	<b>44,81</b>	<b>18,72</b>	<b>19,34</b>	<b>19,21</b>
rc106	8	2	669,43	669,43	669,43	669,43	<b>50,36</b>	<b>25,64</b>	<b>23,47</b>	<b>22,49</b>
rc106	9	2	669,43	669,43	669,43	669,43	<b>65,09</b>	<b>26,59</b>	<b>23,39</b>	<b>27,85</b>
rc107	1	3	1101,80	1101,80	1101,80	1101,80	<b>2,66</b>	<b>2,74</b>	<b>8,11</b>	<b>2,54</b>
rc107	2	2	794,80	794,80	794,80	794,80	<b>27,31</b>	<b>20,89</b>	<b>120,11</b>	<b>23,25</b>
rc107	3	2	788,53	788,53	788,53	788,53	<b>100,02</b>	<b>69,76</b>	<b>106,16</b>	<b>66,44</b>
rc107	4	2	710,91	710,91	710,91	710,91	<b>116,42</b>	<b>97,94</b>	<b>129,70</b>	<b>67,75</b>

Probl.	Pat.	Vehicle Limit	Lower Bound				Computational Time (sec)			
			FULL	CLON	UNI	PARA	FULL	CLON	UNIF	PARA
rc107	5	2	608,58	608,58	608,58	608,58	<b>220,61</b>	<b>191,34</b>	<b>224,02</b>	<b>125,61</b>
rc107	6	2	608,58	608,58	608,58	608,58	<b>392,10</b>	<b>172,52</b>	<b>134,12</b>	<b>200,84</b>
rc107	7	2	608,58	608,58	608,58	608,58	<b>208,82</b>	<b>151,71</b>	<b>118,33</b>	<b>94,95</b>
rc107	8	2	608,58	608,58	608,58	608,58	<b>390,99</b>	<b>126,47</b>	<b>98,14</b>	<b>182,68</b>
rc107	9	2	608,58	608,58	608,58	608,58	<b>316,26</b>	<b>120,57</b>	<b>90,44</b>	<b>146,53</b>
rc108	1	3	1066,30	1066,30	1066,30	1066,30	<b>4,16</b>	<b>4,03</b>	<b>22,11</b>	<b>3,33</b>
rc108	2	2	769,40	769,40	769,40	769,40	<b>122,46</b>	<b>137,55</b>	<b>204,90</b>	<b>79,98</b>
rc108	3	2	745,43	745,43	745,43	745,43	<b>440,60</b>	<b>277,22</b>	<b>382,05</b>	<b>334,54</b>
rc108	4	2	656,07	656,07	656,07	656,07	<b>803,96</b>	<b>360,85</b>	<b>665,76</b>	<b>536,46</b>
rc108	5	2	546,17	546,17	N/A	546,17	<b>808,90</b>	<b>933,02</b>	<b>1404,46</b>	<b>561,87</b>
rc108	6	2	546,17	546,17	546,17	546,17	<b>1078,89</b>	<b>761,67</b>	<b>606,76</b>	<b>664,92</b>
rc108	7	2	546,17	546,17	546,17	546,17	<b>1255,08</b>	<b>712,74</b>	<b>620,02</b>	<b>737,28</b>
rc108	8	2	546,17	546,17	546,17	546,17	<b>2257,79</b>	<b>705,94</b>	<b>591,32</b>	<b>1351,21</b>
rc108	9	2	546,17	546,17	546,17	546,17	<b>1928,54</b>	<b>590,74</b>	<b>462,31</b>	<b>1107,33</b>

\* LDS is not used.

## APPENDIX B: BRANCH-AND-PRICE RESULTS

We present here the original detailed results, which were summarized in Chapter 5.

### APPENDIX B.1: BRANCH-AND-PRICE RESULTS

Table B.1 presents the results for the instances solved using B&P, and employs both branching methods (2br and P+1). Specifically, Table B.1 provides the problem ID, the problem set, the period window pattern (1 to 9), the maximum allowable number of vehicles per period and, for each method, the cost of the integer solution, the total nodes created, the nodes explored, the number of nodes explored since the best integer solution was initially reached, and the time needed to calculate the integer solution. The analysis of these results is presented in Section 5.4.3.

Table B.1: Results on  $P + 1$  and  $2br$  methods for obtaining integer solutions

No	Prob. Set	Pat.	Max Veh.	P+1 method				2br method					
				IB	Total Nodes	Nodes Explored	First Occ.(1)	IB Time	IB	Total Nodes	Nodes Explored	First Occ.(1)	IB Time
1	r101	1	6	1549.5	11	9	8	3.145	1549.5	11	9	8	4.219
3	r101	3	4	1191.7	5	3	3	5.086	1191.7	5	3	3	4.986
4	r101	4	4	1157.8	71	63	50	115.128	1157.8	71	63	50	114.49
6	r101	6	4	1049	10	4	4	18.432	1049	9	7	4	29.608
7	r101	7	4	1049	5	5	2	26.483	1049	7	7	2	33.462
8	r101	8	4	1049	6	6	2	31.72	1049	9	9	2	46.334
9	r101	9	4	1049	13	13	3	74.073	1049	23	23	2	134.524
11	r102	2	3	1183.9	3	3	3	7.843	1183.9	3	3	3	7.539
12	r102	3	3	1107.7	9	9	2	39.883	1107.7	9	9	2	40.468
13	r102	4	3	1051.7	5	3	3	27.463	1051.7	5	3	3	27.007

No	Prob. Set	Pat.	Max Veh.	P+1 method				2br method					
				IB	Total Nodes	Nodes Explored	First Occ.(1)	IB Time	IB	Total Nodes	Nodes Explored	First Occ.(1)	IB Time
14	r102	5	3	1040.1	15	9	8	93.641	1040.1	15	9	8	93.119
15	r102	6	3	924.6	7	7	2	75.501	924.6	11	11	2	104.641
19	r103	1	4	1278.5	9	9	4	7.334	1278.5	9	9	4	7.196
21	r103	3	3	967.8	7	5	5	54.479	967.8	7	5	5	55.144
22	r103	4	3	872.7	7	5	4	99.714	872.7	7	5	4	107.084
23	r103	5	3	862.8	34	24	19	647.658	862.8	31	23	18	611.412
24	r103	6	3	786.9	10	4	4	149.489	786.9	9	5	5	171.566
25	r103	7	3	777.9	73	41	18	1145.506	777.9	25	19	8	533.84
26	r103	8	3	777.9	90	26	17	813.423	777.9	65	41	26	1250.576
27	r103	9	3	777.9	121	31	20	1091.584	777.9	119	77	46	2959.26
28	r104	1	2	1126.5	7	7	4	15.374	1126.5	7	7	4	14.07
30	r104	3	2	794.5	44	42	5	1621.897	794.5	23	23	4	894.932
38	r105	2	3	1168.3	25	19	19	23.243	1168.3	23	17	17	22.86
39	r105	3	3	1082.1	13	7	7	20.486	1082.1	11	9	4	24.41
40	r105	4	3	1040.2	70	59	40	205.51	1040.2	71	65	58	230.134
41	r105	5	3	1026.7	195	170	120	690.884	1026.7	153	121	84	465.756
42	r105	6	3	914.2	686	531	192	2769.084	914.2	721	653	272	3059.804
43	r105	7	3	904.3	261	145	98	914.757	904.3	227	199	54	1040.486
46	r106	1	4	1320.8	3	3	3	2.804	1320.8	3	3	3	2.56
47	r106	2	3	1046	5	5	2	18.318	1046	5	5	2	17.661
48	r106	3	3	976	500	470	447	1880.641	976	441	433	94	1637.895
49	r106	4	3	880.4	13	9	6	108.778	880.4	13	9	6	109.605
50	r106	5	3	865.1	168	83	75	1101.317	865.1	63	49	30	664.597
51	r106	6	3	798	4	4	4	90.385	798	5	5	5	105.669
52	r106	7	3	798	5	5	5	109.338	798	15	15	8	284.421

No	Prob. Set	Pat.	Max Veh.	P+1 method				2br method					
				IB	Total Nodes	Nodes Explored	First Occ.(1)	IB Time	IB	Total Nodes	Nodes Explored	First Occ.(1)	IB Time
53	r106	8	3	798	31	11	11	225.356	798	25	23	12	442.995
54	r106	9	3	798	43	13	13	317.954	798	23	23	12	523.242
55	r107	1	4	1221.4	3	3	3	3.524	1221.4	3	3	3	3.588
56	r107	2	3	946.7	3	3	3	23.39	946.7	3	3	3	22.5
57	r107	3	3	878.9	41	26	22	300.741	878.9	29	21	21	256.736
58	r107	4	3	800.1	49	41	12	845.913	800.1	35	27	25	482.529
59	r107	5	3	768.6	87	45	32	2001.875	768.6	55	43	24	2009.306
64	r108	1	3	1126.5	33	19	16	31.039	1126.5	33	19	16	31.988
66	r108	3	3	752.2	56	31	26	752.168	752.2	47	29	24	708.455
74	r109	2	3	1005.6	17	13	9	37.992	1005.6	11	9	4	25.275
75	r109	3	3	915.9	30	28	9	144.322	915.9	35	33	19	158.254
76	r109	4	3	842.5	92	87	50	603.437	842.5	101	97	44	627.876
77	r109	5	3	825.5	154	75	64	662.644	825.5	55	41	26	352.565
82	r110	1	3	1242.7	23	13	12	11.777	1242.7	23	13	12	11.099
84	r110	3	2	905	13	9	8	145.019	905	15	11	9	168.402
88	r110	7	2	702	43	21	13	728.616	702	27	19	15	512.715
91	r111	1	3	1216.6	7	7	4	8.783	1216.6	7	7	4	8.695
92	r111	2	3	926.7	148	125	63	588.167	926.7	165	151	62	648.799
93	r111	3	3	861.4	66	49	38	434.434	861.4	79	63	47	558.595
94	r111	4	3	769.5	68	42	37	782.906	769.5	19	17	6	327.117
101	r112	2	3	817.9	9	7	5	54.305	817.9	9	7	5	54.113
102	r112	3	3	751.7	5	5	2	113.828	751.7	5	5	2	112.588
127	c103	1	2	451.6	5	3	3	12.152	451.6	5	3	3	12.014
172	c108	1	2	453.7	9	5	5	10.048	453.7	9	5	5	9.917
181	c109	1	2	449.1	85	65	45	81.733	449.1	85	65	45	78.478

No	Prob. Set	Pat.	Max Veh.	P+1 method				2br method					
				IB	Total Nodes	Nodes Explored	First Occ.(1)	IB Time	IB	Total Nodes	Nodes Explored	First Occ.(1)	IB Time
190	rc101	1	4	1399.8	241	241	22	82.268	1399.8	241	241	22	80.969
199	rc102	1	3	1267.4	49	49	24	29.224	1267.4	49	49	24	28.368
218	rc104	2	2	781	24	22	10	756.836	781	29	23	16	773.189
236	rc106	2	2	892.1	15	11	7	53.299	892.1	25	23	7	108.359
246	rc107	3	2	793.5	28	22	5	1021.212	793.5	35	33	12	1317.692
254	rc108	2	2	770.6	11	6	6	431.697	770.6	9	9	9	491.354

(1) Number of nodes explored since the best integer solution initially reached

## APPENDIX B.2: BRANCH-AND-PRICE HEURISTIC PRUNING RESULTS

Table B.2 presents the results obtained by using the heuristic pruning technique with the  $P + 1$  branching method for different values of the parameter  $\lambda$  ( $\lambda = 1.00, 0.95, 0.90, 0.80, 0.75, 0.50$ ). Specifically, Table B.2 presents the problem ID, the problem set, the period window pattern (1 to 9), the maximum allowable number of vehicles per period, and, for each method, the cost of the integer solution, and its deviation from the cost of the optimal solution (%), the number of nodes explored, the number of nodes explored since the best integer solution was initially reached, and the time required to calculate the integer solution. The analysis of these results is presented in Section 5.4.4.

Table B.2: Pruning heuristic results ( $P + 1$  B&P method)

#	Prob. Set	Pat.	Max Veh.	IB	Cost Deviation							Nodes Explored							First Occurrence <sup>(1)</sup>							IB Computational Time						
					1.00	0.95	0.90	0.85	0.80	0.75	0.50	1.00	0.95	0.90	0.85	0.80	0.75	0.50	1.00	0.95	0.90	0.85	0.80	0.75	0.50	1.00	0.95	0.90	0.85	0.80	0.75	0.50
3	r101	3	4	1191.7	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	3	3	3	3	3	3	1	1	1	1	1	1	1	3.0	5.4	4.9	5.4	5.1	5.2	5.2
4	r101	4	4	1157.8	0.46%	0.16%	0.16%	0.16%	0.16%	0.16%	0.00%	1	9	9	9	15	17	27	1	7	7	7	13	13	11	4.0	20.3	20.5	20.0	34.0	36.0	52.8
6	r101	6	4	1049	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	4	4	4	4	4	4	1	1	1	1	1	1	1	5.9	17.3	16.9	18.6	17.8	16.8	17.9
7	r101	7	4	1049	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	5	5	5	5	5	5	1	1	1	1	1	1	1	6.3	26.1	25.2	27.1	27.2	26.2	25.7
8	r101	8	4	1049	0.26%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	6	6	6	6	6	6	1	2	2	2	2	2	2	6.6	31.9	32.2	33.8	33.6	33.3	31.2
9	r101	9	4	1049	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	7	7	7	7	7	7	1	1	1	1	1	1	1	7.9	47.4	43.9	46.2	47.2	44.1	44.1
11	r102	2	3	1183.9	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	3	3	3	3	3	3	1	1	1	1	1	1	1	4.3	7.9	8.0	12.1	7.9	8.1	8.2
12	r102	3	3	1107.7	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	3	3	3	3	3	5	1	1	1	1	1	1	1	7.9	18.4	18.0	18.6	18.8	18.3	26.2
13	r102	4	3	1051.7	0.10%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	3	3	3	3	3	3	1	3	3	3	3	3	3	16.2	28.7	26.9	27.0	27.8	27.3	27.3
14	r102	5	3	1040.1	0.05%	0.05%	0.05%	0.05%	0.05%	0.05%	0.00%	1	7	7	7	7	7	7	1	1	1	1	1	1	6	22.5	68.7	67.3	68.4	68.0	68.2	72.8
15	r102	6	3	924.6	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	4	4	4	4	4	4	1	1	1	1	1	1	1	15.1	43.4	42.8	43.9	43.9	43.3	44.0
19	r103	1	4	1278.5	0.08%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	7	7	7	7	7	7	1	4	4	4	4	4	4	1.6	5.9	5.1	5.5	5.6	5.3	5.3
21	r103	3	3	967.8	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	5	5	5	5	5	5	1	1	1	1	1	1	1	16.9	55.1	53.7	55.3	54.5	54.5	55.3
22	r103	4	3	872.7	1.16%	1.04%	0.00%	0.00%	0.00%	0.00%	0.00%	1	5	5	5	5	5	5	1	2	4	4	4	4	4	36.3	95.9	101.4	104.7	103.4	102.3	100.8

#	Prob. Set	Pat.	Max Veh.	IB	Cost Deviation							Nodes Explored							First Occurrence <sup>(1)</sup>							IB Computational Time						
					1.00	0.95	0.90	0.85	0.80	0.75	0.50	1.00	0.95	0.90	0.85	0.80	0.75	0.50	1.00	0.95	0.90	0.85	0.80	0.75	0.50	1.00	0.95	0.90	0.85	0.80	0.75	0.50
23	r103	5	3	862.8	0.13%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	14	14	14	14	14	16	1	6	6	6	6	6	6	69.2	421.0	416.6	423.1	418.6	418.5	457.7
24	r103	6	3	786.9	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	4	4	4	4	4	4	1	1	1	1	1	1	1	66.2	149.6	146.8	150.5	147.9	148.0	148.3
25	r103	7	3	777.9	0.19%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	25	25	25	25	25	29	1	14	14	14	14	14	18	35.0	750.0	745.8	751.9	740.7	748.2	842.5
26	r103	8	3	777.9	0.36%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	21	21	21	21	21	21	1	17	17	17	17	17	17	46.6	664.6	656.1	656.6	653.9	657.4	656.5
27	r103	9	3	777.9	0.19%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	25	25	25	25	25	25	1	20	20	20	20	20	20	36.8	886.7	876.6	872.3	874.9	877.0	878.9
28	r104	1	2	1126.5	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	7	7	7	7	7	7	1	1	1	1	1	1	1	3.6	15.0	14.8	15.3	14.5	15.1	15.0
30	r104	3	2	794.5	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	20	22	22	26	26	29	1	1	1	1	1	1	1	129.6	846.9	907.6	906.2	1033.9	1034.8	1149.6
38	r105	2	3	1168.3	0.01%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	11	11	13	13	13	13	1	2	2	2	2	2	2	2.5	14.6	14.5	17.5	16.9	16.5	16.7
39	r105	3	3	1082.1	0.35%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	7	7	7	7	7	7	1	7	7	7	7	7	7	5.4	20.6	19.9	20.2	20.1	20.4	20.8
40	r105	4	3	1040.2	0.10%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	17	17	19	19	19	25	1	12	12	12	12	12	18	10.0	62.6	61.4	67.7	67.8	67.9	90.0
41	r105	5	3	1026.7	0.21%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	11	13	34	34	42	88	1	2	2	2	2	2	2	11.9	58.6	66.9	148.6	149.3	180.1	369.6
42	r105	6	3	914.2	0.28%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	57	57	57	57	131	177	1	29	29	29	29	41	75	10.6	347.8	341.6	338.4	343.9	761.0	1045.9
43	r105	7	3	904.3	0.83%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	105	105	105	105	105	105	1	6	6	6	6	6	6	9.8	700.2	685.4	685.9	688.3	691.1	691.5
46	r106	1	4	1320.8	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	3	3	3	3	3	3	1	1	1	1	1	1	1	1.4	2.7	2.6	2.6	2.7	2.8	2.6
47	r106	2	3	1046	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	3	3	5	5	5	5	1	1	1	1	1	1	1	5.1	12.5	11.9	17.4	17.4	17.9	17.6
48	r106	3	3	976	0.77%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	31	38	48	48	48	126	1	11	11	11	11	11	43	11.6	157.0	181.4	218.3	222.6	221.7	590.6
49	r106	4	3	880.4	0.37%	0.37%	0.37%	0.37%	0.37%	0.37%	0.00%	1	11	11	11	13	13	7	1	1	1	1	1	1	6	23.0	113.6	112.1	111.4	123.9	124.7	92.8
50	r106	5	3	865.1	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	7	7	7	7	7	17	1	1	1	1	1	1	1	31.8	112.7	112.9	111.4	111.6	113.3	254.3
51	r106	6	3	798	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	4	4	4	4	4	4	1	1	1	1	1	1	1	28.8	90.9	90.9	90.1	90.6	90.4	89.3
52	r106	7	3	798	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	5	5	5	5	5	5	1	1	1	1	1	1	1	25.1	107.1	106.3	106.2	105.8	106.9	106.1
53	r106	8	3	798	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	11	11	11	11	11	11	1	1	1	1	1	1	1	24.5	226.6	225.3	222.7	225.3	223.8	223.1
54	r106	9	3	798	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	13	13	13	13	13	13	1	1	1	1	1	1	1	28.8	316.0	314.9	313.7	315.7	313.4	311.4
55	r107	1	4	1221.4	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	3	3	3	3	3	3	1	1	1	1	1	1	1	2.1	3.6	3.8	3.6	3.7	3.6	3.6
56	r107	2	3	946.7	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	3	3	3	3	3	3	1	1	1	1	1	1	1	11.7	22.9	22.9	22.5	23.4	22.6	23.0
57	r107	3	3	878.9	0.15%	0.15%	0.15%	0.15%	0.15%	0.00%	0.00%	1	11	11	11	13	18	18	1	1	1	1	1	8	8	26.1	151.4	149.2	147.9	175.2	210.3	211.0

#	Prob. Set	Pat.	Max Veh.	IB	Cost Deviation							Nodes Explored							First Occurrence <sup>(1)</sup>							IB Computational Time						
					1.00	0.95	0.90	0.85	0.80	0.75	0.50	1.00	0.95	0.90	0.85	0.80	0.75	0.50	1.00	0.95	0.90	0.85	0.80	0.75	0.50	1.00	0.95	0.90	0.85	0.80	0.75	0.50
58	r107	4	3	800.1	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	5	5	5	5	5	5	1	1	1	1	1	1	1	41.0	126.6	125.3	124.0	125.3	124.8	124.7
59	r107	5	3	768.6	1.47%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	12	19	19	19	19	22	1	11	16	16	16	16	19	93.9	577.6	887.7	885.2	888.8	892.1	982.2
64	r108	1	3	1126.5	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	11	11	11	11	11	11	1	1	1	1	1	1	1	4.0	18.6	18.5	18.5	17.9	17.6	18.5
66	r108	3	3	752.2	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	11	11	11	11	11	11	1	1	1	1	1	1	1	43.5	305.2	302.3	302.8	304.0	300.8	336.3
74	r109	2	3	1005.6	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	7	7	7	7	7	7	1	1	1	1	1	1	1	5.8	21.3	22.0	21.2	22.1	22.3	27.2
75	r109	3	3	915.9	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	10	10	10	14	14	18	1	1	1	1	1	1	1	10.1	52.1	51.2	52.0	73.0	74.6	108.7
76	r109	4	3	842.5	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	18	18	18	18	18	32	1	1	1	1	1	1	1	18.1	146.9	146.4	145.8	146.4	145.4	299.6
77	r109	5	3	825.5	1.50%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	13	13	15	15	18	22	1	6	6	6	6	6	6	25.1	155.8	153.7	173.9	175.1	201.9	267.9
82	r110	1	3	1242.7	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	11	11	11	11	11	13	1	1	1	1	1	1	1	1.8	10.4	10.3	10.2	11.4	10.1	14.9
84	r110	3	2	905	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	9	9	9	9	9	9	1	1	1	1	1	1	1	31.7	148.2	147.5	147.7	145.9	147.0	168.5
88	r110	7	2	702	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	21	21	21	21	21	21	1	1	1	1	1	1	1	42.2	732.5	730.3	733.0	730.2	731.2	806.2
91	r111	1	3	1216.6	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	5	5	5	5	5	5	1	1	1	1	1	1	1	2.1	6.8	6.8	6.5	6.9	7.1	7.0
92	r111	2	3	926.7	0.11%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	23	23	23	31	33	41	1	3	3	3	3	3	3	9.5	119.4	116.6	119.1	156.5	160.9	240.7
93	r111	3	3	861.4	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	10	10	10	10	10	26	1	1	1	1	1	1	1	19.9	94.3	93.7	92.8	96.0	94.2	300.3
94	r111	4	3	769.5	2.07%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	13	13	13	16	18	22	1	4	4	4	4	4	4	50.0	227.4	226.4	224.5	303.0	357.2	486.6
101	r112	2	3	817.9	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	5	5	5	5	5	5	1	1	1	1	1	1	1	13.7	43.4	43.9	43.2	43.2	42.0	48.6
102	r112	3	3	751.7	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	3	3	3	3	3	3	1	1	1	1	1	1	1	35.4	72.3	71.8	71.1	71.7	71.6	76.9
127	c103	1	2	451.6	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	3	3	3	3	3	3	1	1	1	1	1	1	1	5.4	12.3	12.3	12.2	12.7	12.2	12.6
172	c108	1	2	453.7	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	5	5	5	5	5	5	1	1	1	1	1	1	1	3.4	10.0	10.2	9.8	10.2	10.3	10.3
181	c109	1	2	449.1	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	23	23	23	23	23	23	1	1	1	1	1	1	1	4.6	35.1	34.2	34.5	34.9	34.6	41.8
190	rc101	1	4	1399.8	0.05%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	31	59	101	129	157	241	1	18	20	20	22	22	22	1.2	13.5	24.5	36.0	47.0	58.7	103.7
199	rc102	1	3	1267.4	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	9	15	25	37	47	49	1	1	1	1	1	1	1	1.8	6.9	10.7	16.8	23.2	28.1	34.6
218	rc104	2	2	781	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	8	8	8	8	8	10	1	1	1	1	1	1	1	75.0	348.3	342.9	343.0	345.0	344.8	418.6
236	rc106	2	2	892.1	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	5	5	5	5	5	5	1	1	1	1	1	1	1	7.2	30.9	30.3	30.8	30.5	30.4	31.4
246	rc107	3	2	793.5	0.16%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	13	13	13	13	13	13	1	5	5	5	5	5	5	76.9	694.0	684.6	683.1	688.1	684.8	709.4

#	Prob. Set	Pat.	Max Veh.	IB	Cost Deviation							Nodes Explored							First Occurrence <sup>(1)</sup>							IB Computational Time						
					1.00	0.95	0.90	0.85	0.80	0.75	0.50	1.00	0.95	0.90	0.85	0.80	0.75	0.50	1.00	0.95	0.90	0.85	0.80	0.75	0.50	1.00	0.95	0.90	0.85	0.80	0.75	0.50
254	rc108	2	2	770.6	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1	6	6	6	6	6	6	1	1	1	1	1	1	1	97.2	432.9	425.1	424.8	426.5	425.6	438.0

(1) Number of nodes explored since the best integer solution initially reached

## APPENDIX C: DETAILED RESULTS ON THE EXPERIMENTS REGARDING THE PENALTY FUNCTIONS

Table C.1 presents the analytical results for the instances solved in Section 6.3.3 concerning the utilization of different penalty functions in the multi-period vehicle routing setting. Specifically, Table C.1 provides the period window pattern (3 or 5), the planning horizon (1 to 5), and the routing cost (1), total cost (incl. penalties) (2), the number of customer orders considered in each MPVRPTW, the number of routed customer orders (3), and the number of unserved customer orders (4) within the planning horizon, as well as the unit routing cost per served customer (5) for each one of the penalty functions ( $\gamma = 1, 2, 3, 4$  and 5).

Table C.1: Pruning heuristic results ( $P + 1$  B&P method)

Pattern	Planning Horizon																									
		$\gamma = 1$ (flat)					$\gamma = 2$ (step)					$\gamma = 3$ (square)					$\gamma = 4$ (quad)					$\gamma = 5$ (linear)				
		(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
		R101																								
3	1	830,1	1654,7	34	16	24,41	1053,4	1671,2	38	12	27,72	1069,0	1590,1	39	11	27,41	1069,0	1590,1	39	11	27,41	1069,0	1590,1	39	11	27,41
3	2	901,3	1508,5	38	12	23,72	1048,0	1619,5	39	11	26,87	1048,0	1619,5	39	11	26,87	1048,0	1619,5	39	11	26,87	1004,7	1557,3	40	10	25,12
3	3	949,4	1480,2	40	10	23,74	1041,6	1621,9	39	11	26,71	1041,6	1621,9	39	11	26,71	1041,6	1621,9	39	11	26,71	992,3	1532,5	40	10	24,81
3	4	1004,1	1503,6	41	9	24,49	1035,6	1631,9	39	11	26,55	1041,6	1621,9	39	11	26,71	1041,6	1621,9	39	11	26,71	992,3	1532,5	40	10	24,81
3	5	1002,1	1483,0	41	9	24,44	1033,6	1557,2	40	10	25,84	1041,6	1621,9	39	11	26,71	1041,6	1621,9	39	11	26,71	992,3	1532,5	40	10	24,81
		R101																								
5	1	810,8	1655,8	34	16	23,85	1059,6	1635,7	39	11	27,17	1069,0	1590,1	39	11	27,41	1069,0	1590,1	39	11	27,41	1013,9	1514,3	40	10	25,35
5	2	839,3	1546,6	37	13	22,68	1014,6	1624,1	39	11	26,02	1048,0	1619,5	39	11	26,87	1048,0	1619,5	39	11	26,87	1004,7	1557,3	40	10	25,12
5	3	893,3	1514,3	38	12	23,51	1040,9	1612,4	39	11	26,69	1042,8	1623,1	39	11	26,74	1042,8	1623,1	39	11	26,74	991,9	1482,1	40	10	24,80
5	4	969,0	1427,4	40	10	24,23	996,4	1535,7	39	11	25,55	997,4	1615,7	39	11	25,57	997,4	1615,7	39	11	25,57	945,4	1490,6	40	10	23,64
5	5	964,4	1414,3	41	9	23,52	1001,6	1547,4	40	10	25,04	997,4	1615,7	39	11	25,57	997,4	1615,7	39	11	25,57	945,4	1490,6	40	10	23,64

Pattern	Planning Horizon	$\gamma = 1$ (flat)					$\gamma = 2$ (step)					$\gamma = 3$ (square)					$\gamma = 4$ (quad)					$\gamma = 5$ (linear)				
		(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
		R102																								
3	1	992,4	1460,4	42	8	23,63	1134,8	1445,3	45	5	25,22	1061,3	1308,8	45	5	23,58	1061,3	1308,8	45	5	23,58	1016,3	1500,0	43	7	23,63
3	2	950,6	1356,9	43	7	22,11	1089,5	1292,9	46	4	23,68	1071,5	1292,7	46	4	23,29	1071,5	1292,7	46	4	23,29	1072,5	1234,2	47	3	22,82
3	3	1055,6	1217,3	47	3	22,46	1091,1	1262,8	47	3	23,21	1087,1	1258,8	47	3	23,13	1091,1	1262,8	47	3	23,21	1087,1	1258,8	47	3	23,13
3	4	1033,4	1194,3	47	3	21,99	1087,1	1258,8	47	3	23,13	1087,1	1258,8	47	3	23,13	1087,1	1258,8	47	3	23,13	1087,1	1258,8	47	3	23,13
3	5	1033,4	1194,3	47	3	21,99	1087,1	1258,8	47	3	23,13	1087,1	1258,8	47	3	23,13	1087,1	1258,8	47	3	23,13	1087,1	1258,8	47	3	23,13
		R105																								
3	1	906,7	1505,4	39	11	23,25	1167,0	1469,1	45	5	25,93	1307,2	1486,4	46	4	28,42	1307,2	1486,4	46	4	28,42	1289,9	1458,3	46	4	28,04
3	2	935,1	1305,2	43	7	21,75	1147,2	1483,3	45	5	25,49	1268,9	1373,8	47	3	27,00	1268,9	1373,8	47	3	27,00	1232,2	1380,5	47	3	26,22
3	3	904,1	1255,0	43	7	21,03	1095,5	1427,2	45	5	24,34	1263,7	1377,4	47	3	26,89	1263,7	1377,4	47	3	26,89	1263,7	1377,4	47	3	26,89
3	4	1047,3	1268,3	46	4	22,77	1169,2	1437,4	46	4	25,42	1255,8	1389,9	47	3	26,72	1255,8	1389,9	47	3	26,72	1255,8	1389,9	47	3	26,72
3	5	1040,0	1262,2	46	4	22,61	1044,6	1299,3	45	5	23,21	1255,8	1389,9	47	3	26,72	1255,8	1389,9	47	3	26,72	1255,8	1389,9	47	3	26,72
		R105																								
5	1	842,9	1494,8	37	13	22,78	1311,2	1519,6	46	4	28,50	1307,2	1486,4	46	4	28,42	1307,2	1486,4	46	4	28,42	1016,6	1325,8	44	6	23,10
5	2	837,8	1359,5	41	9	20,43	1258,8	1464,8	46	4	27,37	1267,4	1401,5	47	3	26,97	1267,4	1401,5	47	3	26,97	1152,5	1313,8	47	3	24,52
5	3	950,0	1185,9	45	5	21,11	1084,8	1369,7	45	5	24,11	1204,9	1347,8	47	3	25,64	1204,9	1347,8	47	3	25,64	1204,9	1347,8	47	3	25,64
5	4	1001,4	1201,2	46	4	21,77	1148,6	1376,4	46	4	24,97	1199,7	1342,6	47	3	25,53	1199,7	1342,6	47	3	25,53	1199,7	1342,6	47	3	25,53
5	5	1001,4	1201,2	46	4	21,77	1054,2	1228,8	46	4	22,92	1199,7	1342,6	47	3	25,53	1199,7	1342,6	47	3	25,53	1199,7	1342,6	47	3	25,53
		R109																								
3	1	890,5	1293,1	42	8	21,20	1061,1	1091,2	49	1	21,66	1127,7	1127,7	50	0	22,55	1127,7	1127,7	50	0	22,55	1021,8	1072,7	49	1	20,85
3	2	795,2	1223,4	42	8	18,93	1119,6	1119,6	50	0	22,39	1121,5	1121,5	50	0	22,43	1121,5	1121,5	50	0	22,43	1044,8	1095,7	49	1	21,32
3	3	861,7	1070,5	46	4	18,73	1119,6	1119,6	50	0	22,39	1106,7	1106,7	50	0	22,13	1106,7	1106,7	50	0	22,13	1106,7	1106,7	50	0	22,13
3	4	951,7	1002,6	49	1	19,42	1101,0	1101,0	50	0	22,02	1106,7	1106,7	50	0	22,13	1106,7	1106,7	50	0	22,13	1106,7	1106,7	50	0	22,13
3	5	951,7	1002,6	49	1	19,42	1083,7	1083,7	50	0	21,67	1083,7	1083,7	50	0	21,67	1083,7	1083,7	50	0	21,67	1083,7	1083,7	50	0	21,67

Pattern	Planning Horizon	$\gamma = 1$ (flat)					$\gamma = 2$ (step)					$\gamma = 3$ (square)					$\gamma = 4$ (quad)					$\gamma = 5$ (linear)				
		(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
R110																										
3	1	952,2	1174,7	45	5	21,16	1172,6	1172,6	50	0	23,45	1087,0	1087,0	50	0	21,74	1087,0	1087,0	50	0	21,74	1087,0	1087,0	50	0	21,74
3	2	881,0	1021,5	47	3	18,74	1002,4	1002,4	50	0	20,05	1069,0	1069,0	50	0	21,38	1069,0	1069,0	50	0	21,38	1069,0	1069,0	50	0	21,38
3	3	837,3	946,1	48	2	17,44	902,8	902,8	50	0	18,06	914,3	914,3	50	0	18,29	915,4	915,4	50	0	18,31	914,3	914,3	50	0	18,29
3	4	900,0	900,0	50	0	18,00	900,0	900,0	50	0	18,00	900,0	900,0	50	0	18,00	900,0	900,0	50	0	18,00	900,0	900,0	50	0	18,00
3	5	900,0	900,0	50	0	18,00	900,0	900,0	50	0	18,00	900,0	900,0	50	0	18,00	900,0	900,0	50	0	18,00	900,0	900,0	50	0	18,00
C101																										
3	1	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	513,8	513,8	50	0	10,28	513,8	513,8	50	0	10,28	513,8	513,8	50	0	10,28
3	2	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	3	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	4	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	5	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
C101																										
5	1	327,6	912,4	42	8	7,80	492,9	492,9	50	0	9,86	566,6	566,6	50	0	11,33	566,6	566,6	50	0	11,33	566,6	566,6	50	0	11,33
5	2	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	423,4	423,4	50	0	8,47	398,8	398,8	50	0	7,98	433,6	433,6	50	0	8,67
5	3	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
5	4	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
5	5	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
C105																										
3	1	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	446,1	446,1	50	0	8,92	446,1	446,1	50	0	8,92	446,1	446,1	50	0	8,92
3	2	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	3	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	4	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	5	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25

Pattern	Planning Horizon	$\gamma = 1$ (flat)					$\gamma = 2$ (step)					$\gamma = 3$ (square)					$\gamma = 4$ (quad)					$\gamma = 5$ (linear)				
		(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
C105																										
5	1	327,6	912,4	42	8	7,80	467,8	467,8	50	0	9,36	502,8	502,8	50	0	10,06	502,8	502,8	50	0	10,06	502,8	502,8	50	0	10,06
5	2	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	408,7	408,7	50	0	8,17	398,3	398,3	50	0	7,97	408,7	408,7	50	0	8,17
5	3	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
5	4	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
5	5	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
C106																										
3	1	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	469,9	469,9	50	0	9,40	469,9	469,9	50	0	9,40	469,9	469,9	50	0	9,40
3	2	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	3	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	4	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	5	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
C106																										
5	1	327,6	912,4	42	8	7,80	467,8	467,8	50	0	9,36	596,7	596,7	50	0	11,93	596,7	596,7	50	0	11,93	607,1	607,1	50	0	12,14
5	2	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	433,6	433,6	50	0	8,67	362,4	362,4	50	0	7,25	433,6	433,6	50	0	8,67
5	3	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
5	4	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
5	5	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
C107																										
3	1	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	445,9	445,9	50	0	8,92	445,9	445,9	50	0	8,92	445,9	445,9	50	0	8,92
3	2	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	3	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	4	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	5	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25

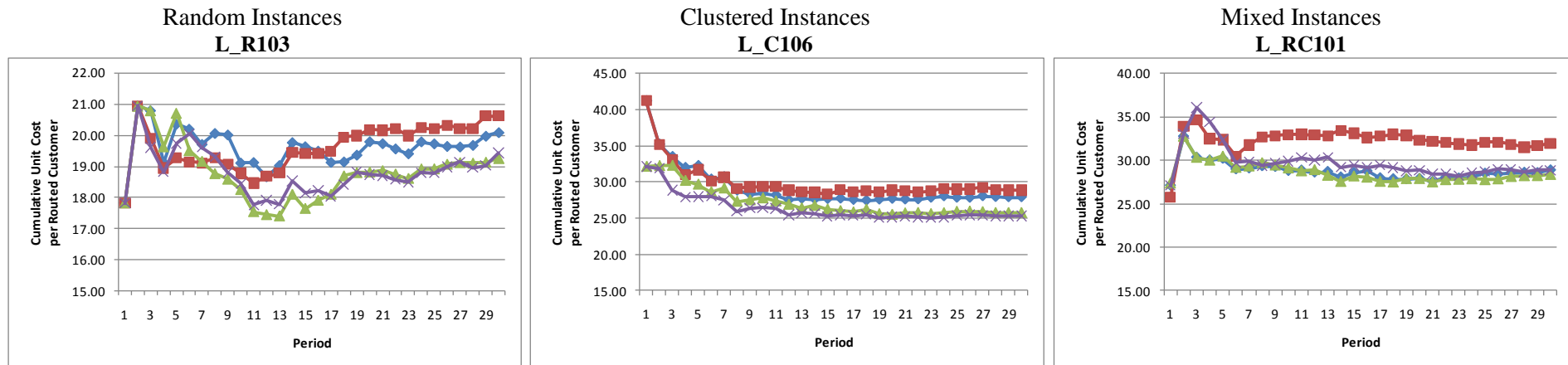
Pattern	Planning Horizon																									
		$\gamma = 1$ (flat)					$\gamma = 2$ (step)					$\gamma = 3$ (square)					$\gamma = 4$ (quad)					$\gamma = 5$ (linear)				
		(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
C107																										
5	1	498,3	498,3	50	0	9,97	498,3	498,3	50	0	9,97	534,6	534,6	50	0	10,69	534,6	534,6	50	0	10,69	534,6	534,6	50	0	10,69
5	2	417,8	417,8	50	0	8,36	397,9	397,9	50	0	7,96	504,1	504,1	50	0	10,08	509,0	509,0	50	0	10,18	504,1	504,1	50	0	10,08
5	3	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
5	4	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
5	5	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
C108																										
3	1	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	452,7	452,7	50	0	9,05	452,7	452,7	50	0	9,05	452,7	452,7	50	0	9,05
3	2	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	3	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	4	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25
3	5	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25	362,4	362,4	50	0	7,25

## APPENDIX D: DETAILED RESULTS OF THE EXPERIMENTAL INVESTIGATION PRESENTED IN CHAPTER 6

We present here the detailed results of the experimental investigation of Chapter 6. These results were summarized in Section 6.4.

### APPENDIX D.1: RESULTS FOR THE QUASI-STATIC INSTANCES

Figure D.1 illustrates the cost per routed customer as it changes over the periods of the long term horizon. These figures correspond to the analysis presented in Section 6.4.1 regarding the quasi-static test instances. The cost value per customer at a certain period is the ratio of the total routing costs for all periods till the period under consideration, over the total number of customers routed till the said period (cumulative unit cost per customer). Results are presented per each combination of planning horizon ( $P$ ) and implementation horizon ( $M$ ). Each Figure presents 3 graphs, one per each instance of the random, clustered and mixed instances tested, respectively.



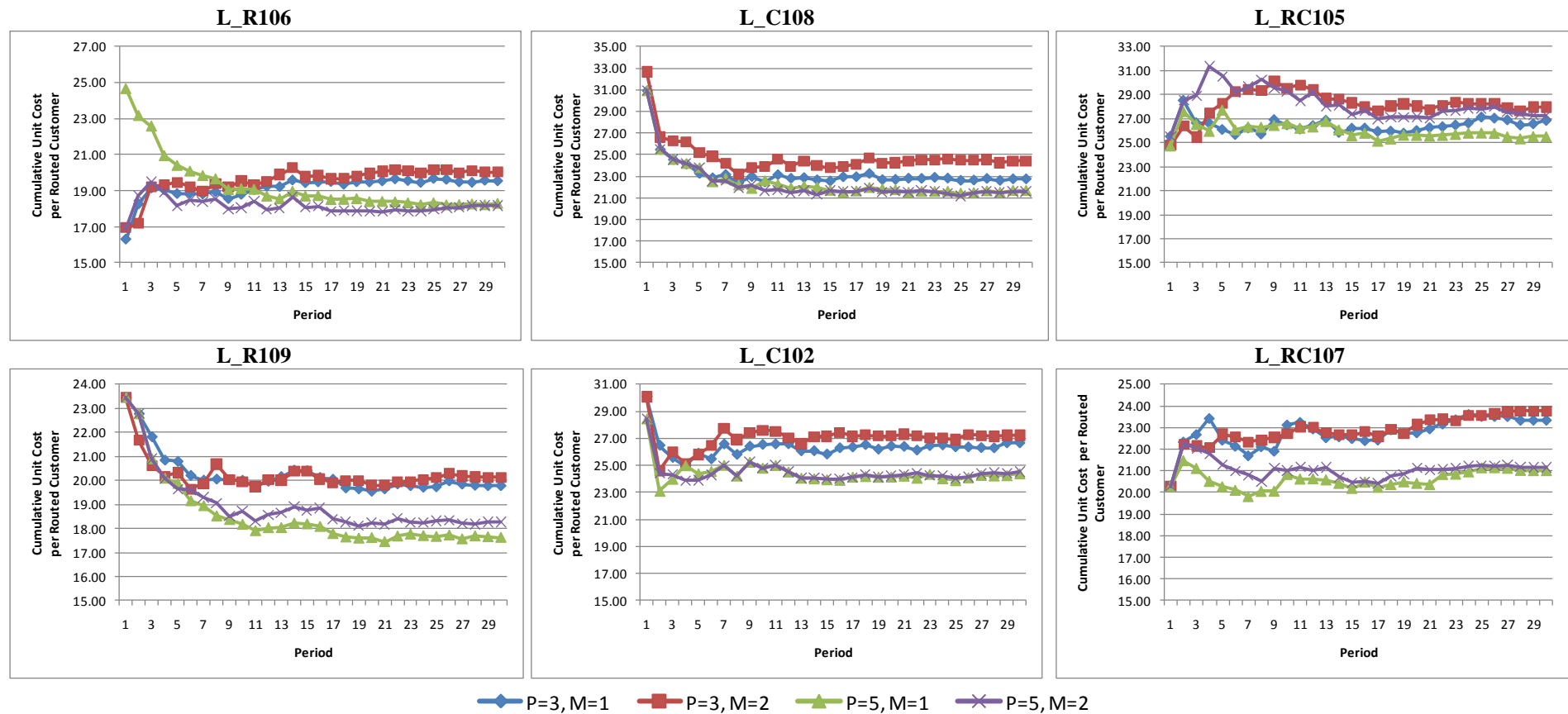
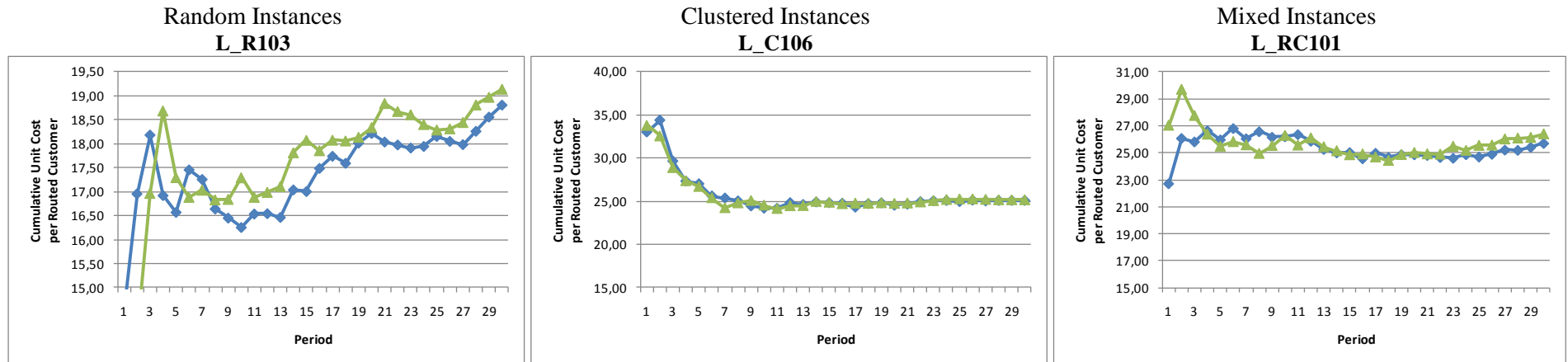


Figure D.1: Cumulative unit cost per routed customer and period for all instances of Section 6.4.1

## APPENDIX D.2: RESULTS FOR THE DYNAMIC INSTANCES OF SECTION 6.4.2 (A)

Figures D.2 illustrates the cost per routed customer as it changes over the periods of the scheduling horizon. These figures correspond to the analysis presented in Section 6.4.2 regarding the dynamic test instances with moderate planning horizon (3 and 5). The cost value per customer at a certain period is the ratio of the total routing costs for all periods till the period under consideration, over the total number of customers routed till the said period (cumulative unit cost per customer). Results are presented per each planning horizon ( $P$ ). Each Figure presents 3 graphs, one per each instance of the random, clustered and mixed instances tested, respectively.



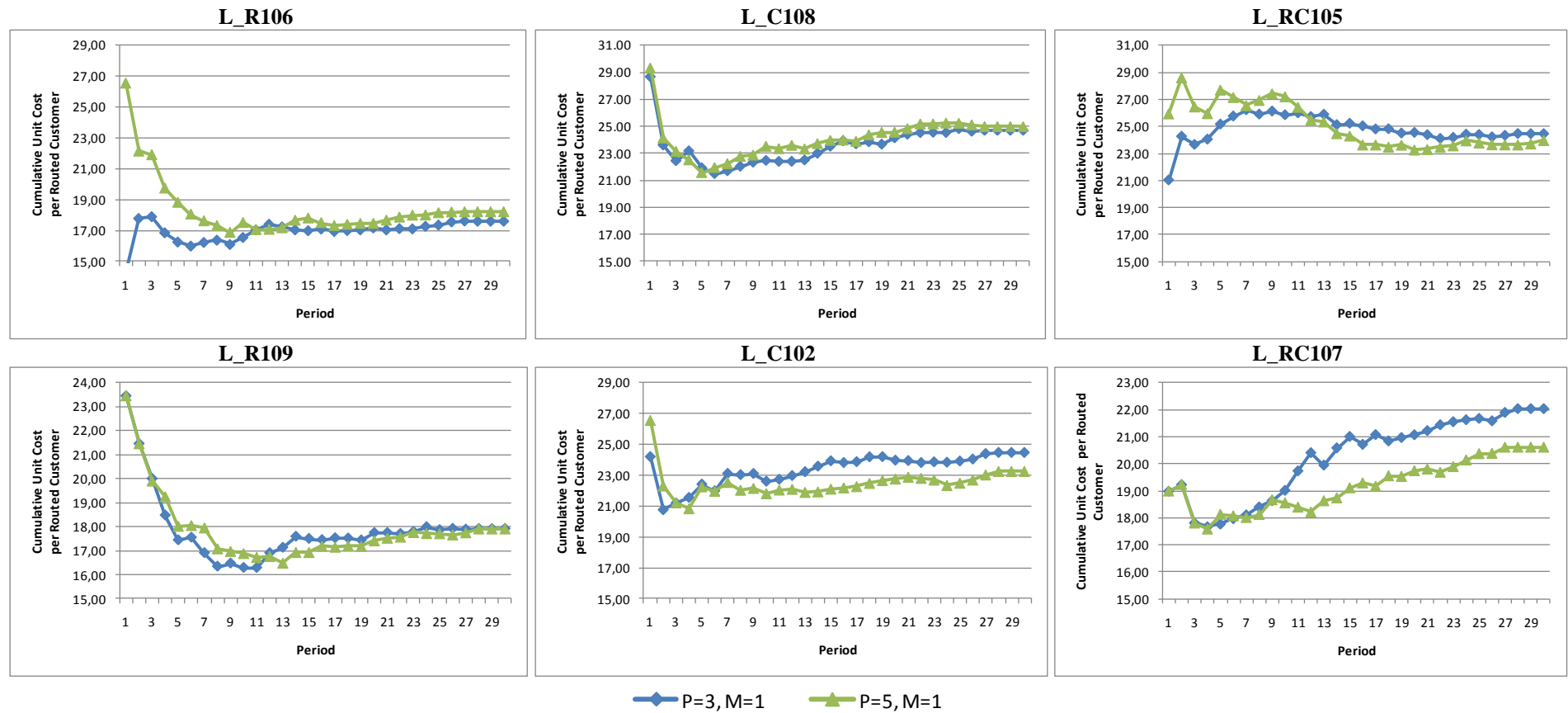


Figure D.2: Cumulative unit cost per routed customer and period for all instances of Section 6.4.2

### APPENDIX D.3: RESULTS OF THE DYNAMIC INSTANCES OF SECTION 6.4.2 (B)

Table D.1 presents the results obtained for the dynamic test instances of Section 6.4.2 for all planning horizons (1 and 7) and for period window pattern #7. The Table presents the instance name, the value of the planning horizon used ( $P$ ), the total number of served customers, and the average routing cost ratio over the 30- period horizon.

Table D.1: Comparative results using different planning horizons (dynamic arrival of customers)

Problem	P	Served Customers	Routing Cost Ratio
<b>L_r103</b>	<b>1</b>	350	21.3
	<b>2</b>	356	19.1
	<b>3</b>	353	18.2
	<b>4*</b>	197	15.5
	<b>5</b>	349	15.9
	<b>6</b>	349	15.5
	<b>7*</b>	307	15.8
<b>L_r106</b>	<b>1</b>	360	21.5
	<b>2</b>	360	19.4
	<b>3</b>	359	16.4
	<b>4</b>	359	15.6
	<b>5</b>	359	15.5
	<b>6</b>	359	15.7
	<b>7</b>	359	16.4
<b>L_r109</b>	<b>1</b>	360	20.9
	<b>2</b>	360	18.9
	<b>3</b>	360	16.0
	<b>4</b>	360	15.5
	<b>5</b>	360	14.6
	<b>6</b>	360	16.2
	<b>7</b>	360	15.6
<b>L_c106</b>	<b>1</b>	360	37.7
	<b>2</b>	360	26.9
	<b>3</b>	360	22.6
	<b>4</b>	360	20.6
	<b>5</b>	360	21.4
	<b>6</b>	360	20.5
	<b>7</b>	360	21.1
<b>L_c108</b>	<b>1</b>	360	32.4
	<b>2</b>	360	23.5
	<b>3</b>	360	21.8
	<b>4</b>	360	20.9
	<b>5</b>	360	20.7
	<b>6</b>	360	21.1
	<b>7</b>	360	22.6

Problem	P	Served Customers	Routing Cost Ratio
<b>L_c102</b>	<b>1</b>	360	34.3
	<b>2</b>	360	26.3
	<b>3</b>	359	23.0
	<b>4</b>	356	21.0
	<b>5</b>	358	20.2
	<b>6</b>	355	20.0
	<b>7</b>	358	21.3
<b>L_rc101</b>	<b>1</b>	337	24.7
	<b>2</b>	340	24.0
	<b>3</b>	332	23.8
	<b>4</b>	335	22.2
	<b>5</b>	336	22.5
	<b>6</b>	337	20.5
	<b>7</b>	326	23.6
<b>L_rc105</b>	<b>1</b>	344	24.2
	<b>2</b>	349	23.8
	<b>3</b>	349	22.0
	<b>4</b>	351	20.5
	<b>5</b>	346	19.5
	<b>6</b>	349	18.9
	<b>7</b>	346	18.5
<b>L_rc107</b>	<b>1</b>	360	23.7
	<b>2</b>	360	20.7
	<b>3</b>	360	19.0
	<b>4</b>	360	19.1
	<b>5</b>	360	18.2
	<b>6</b>	360	18.9
	<b>7</b>	360	18.5

\* In this cases the solution procedure terminated prematurely ( $P = 4, period 17, P = 7, period 27$ ) due to computational complexity

Figures D.3, D.4 and D.5 illustrate the average routing cost ratio per each planning horizon, as presented in Table D.1. Each Figure presents 3 graphs, one per each instance of the random, clustered and mixed instances tested, respectively.

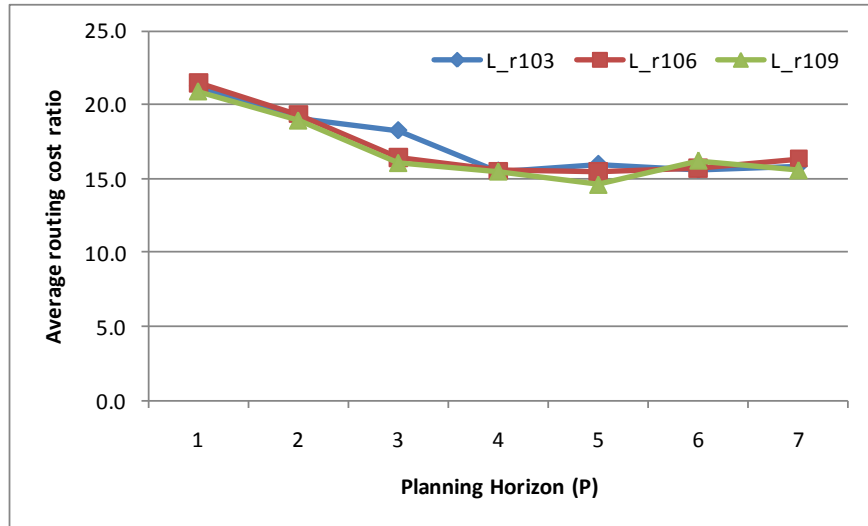


Figure D.3: Cumulative unit cost per routed customer and period (Random Instances)

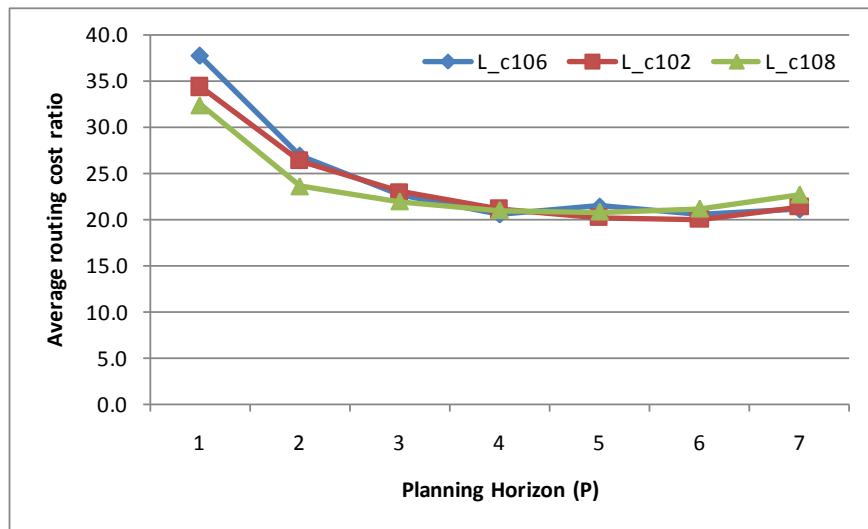


Figure D.4: Cumulative unit cost per routed customer and period (Clustered Instances)

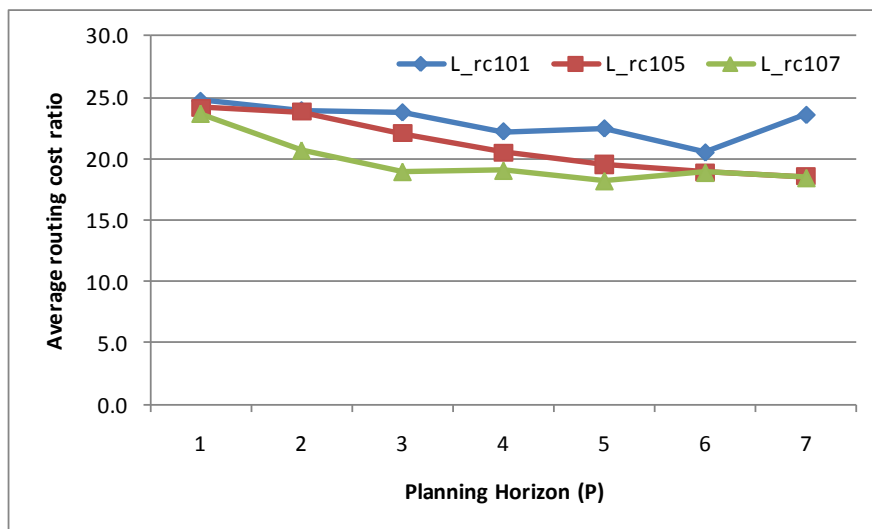


Figure D.5: Cumulative unit cost per routed customer and period (Mixed Instances)

## APPENDIX E: DETAILED RESULTS OF THE EXPERIMENTAL INVESTIGATION PRESENTED IN CHAPTER 7

We present here the detailed results of the experimental investigation of Chapter 7. Table E.1 presents the results obtained for the dynamic test instances of Section 7.3.1 for all planning horizons (1 and 7) and for period window patterns #3, #5 and #7. The Table presents the instance name, the value of the planning horizon used ( $P$ ), the total number of flexible served customers, and the average routing cost ratio (only for the flexible customers) over the 30-period horizon for each one of the period window patterns tested.

Table E.1: Comparative results using different planning horizons (dynamic arrival of customers)

Problem	Planning Horizon	Pattern #3		Pattern #5		Pattern #7	
		Served Customers	Unit Cost per Customer	Served Customers	Unit Cost per Customer	Served Customers	Unit Cost per Customer
L_r103	1	171	13,2	172	12,1	172	12,1
	2	172	7,5	172	6,6	172	6,6
	3	172	6,6	172	4,4	172	4,0
	4			172	4,0	172	3,7
	5			172	4,1	172	2,9
	6					172	2,8
	7					172	2,8
L_r106	1	178	14,4	180	14,2	180	14,0
	2	179	8,6	180	8,0	180	7,7
	3	179	7,5	180	4,6	180	4,4
	4			180	4,4	180	4,0
	5			180	4,4	180	3,8
	6					179	3,8
	7					179	3,5
L_r109	1	179	13,7	180	13,0	180	13,0
	2	180	8,4	180	6,7	180	6,9
	3	180	6,8	180	4,3	180	4,1
	4			180	4,0	180	3,5
	5			180	3,9	180	2,9
	6					180	2,8
	7					180	2,6
L_c106	1	180	20,5	180	20,5	180	20,5
	2	180	10,0	180	8,7	180	8,5
	3	180	8,6	180	6,2	180	5,2
	4			180	4,1	180	3,8
	5			180	4,8	180	3,3
	6					180	3,1
	7					180	3,1

Problem	Planning Horizon	Pattern #3		Pattern #5		Pattern #7	
		Served Customers	Unit Cost per Customer	Served Customers	Unit Cost per Customer	Served Customers	Unit Cost per Customer
L_c102	1	180	20,1	180	20,1	180	20,1
	2	180	10,3	180	9,2	180	9,3
	3	180	8,4	180	5,4	180	4,9
	4			180	4,6	180	4,5
	5			180	4,7	180	3,8
	6					179	3,5
	7					179	3,5
L_c108	1	180	18,1	180	17,8	180	17,8
	2	180	8,1	180	6,8	180	6,6
	3	180	6,8	180	3,9	180	3,8
	4			180	3,5	180	3,1
	5			180	3,2	180	2,7
	6					180	2,4
	7					180	2,4
L_rc101	1	153	13,3	171	13,5	169	12,9
	2	155	12,7	175	11,6	175	10,6
	3	154	11,7	175	11,3	176	9,5
	4			174	7,7	175	5,9
	5			173	8,5	173	5,9
	6					173	5,7
	7					173	4,9
L_rc105	1	159	12,2	173	11,8	174	11,1
	2	162	12,2	174	9,9	174	10,5
	3	159	11,5	174	7,9	174	6,7
	4			174	5,6	174	5,0
	5			174	7,1	173	3,6
	6					173	3,8
	7					173	3,9
L_rc107	1	175	10,6	180	10,4	180	10,3
	2	176	8,4	180	7,0	180	6,8
	3	176	7,7	180	5,6	180	4,3
	4			180	3,8	180	3,0
	5			180	3,7	180	2,6
	6					180	2,5
	7					180	2,4

Table E.2 presents the results obtained for the dynamic test instances of Section 7.3.1 with respect to the penalty cost functions, for all planning horizons (1 and 7) and for period window pattern #7. The Table presents the instance name, the value of the planning horizon used ( $P$ ), the total number of flexible served customers, and the average routing cost ratio

(only for the flexible customers) over the 30-period horizon for each one of the penalty functions ( $\gamma = 1, 2, 3$  and 5).

Table E.2: Comparative results using different penalty cost functions (dynamic arrival of customers)

Problem	Planning Horizon	$\gamma=1$ (flat)		$\gamma=2$ (step)		$\gamma=3$ (square)		$\gamma=3$ (linear)	
		Served Customers	Unit Cost/ Customer	Served Customers	Unit Cost/ Customer	Served Customers	Unit Cost/ Customer	Served Customers	Unit Cost/ Customer
L_r103	1	140	44,9	128	52,1	128	53,7	136	49,1
	2	145	44,3	137	48,7	133	50,8	141	47,1
	3	147	43,0	138	48,0	136	48,8	143	46,7
	4	150	42,3	139	46,8	137	48,6	140	47,4
	5	148	43,0	136	48,8	130	51,9	138	47,8
	6	147	43,8	139	47,4	133	50,3	138	47,7
	7	147	43,3	136	47,8	131	51,3	136	48,8
L_r106	1	142	43,7	141	46,4	130	53,8	140	48,1
	2	148	42,5	141	47,2	137	49,9	141	48,0
	3	148	42,6	140	47,2	137	50,0	142	47,2
	4	154	41,8	139	48,2	138	49,2	145	46,3
	5	152	42,6	140	47,8	139	48,7	144	46,8
	6	151	42,4	138	48,4	134	50,3	142	47,3
	7	148	43,7	134	50,1	134	50,2	141	47,2
L_r109	1	147	42,4	139	47,4	133	52,1	142	47,7
	2	150	41,5	138	47,6	137	49,7	147	45,3
	3	149	41,1	137	49,0	138	49,4	143	47,1
	4	156	40,6	144	46,3	136	49,3	142	46,8
	5	154	41,1	141	46,5	140	47,8	142	47,3
	6	153	41,4	145	45,2	141	46,7	142	46,6
	7	146	43,7	135	49,2	131	51,2	136	48,9
L_c106	1	146	58,6	141	69,8	135	77,7	143	74,0
	2	152	56,5	140	72,5	141	71,9	147	68,8
	3	153	57,5	139	72,6	140	70,9	145	69,7
	4	154	56,4	146	66,4	147	66,6	148	67,2
	5	151	56,8	138	71,0	140	69,2	142	69,9
	6	151	57,4	137	71,3	136	70,9	140	70,6
	7	145	59,8	134	71,3	132	74,7	139	70,3
L_c102	1	140	61,7	133	74,9	133	79,3	134	80,9
	2	143	60,2	133	73,5	136	75,7	137	78,8
	3	145	59,3	132	75,7	136	75,2	136	77,5
	4	149	58,5	137	71,6	137	72,9	139	74,7
	5	143	59,1	132	73,1	132	74,2	135	76,1
	6	143	59,8	132	71,8	133	73,8	133	76,9
	7	139	61,8	128	74,9	129	77,8	130	77,1
L_c108	1	152	56,2	148	65,4	137	74,4	147	68,6
	2	156	55,5	150	65,3	143	69,6	148	67,7

Problem	Planning Horizon	$\gamma=1$ (flat)		$\gamma=2$ (step)		$\gamma=3$ (square)		$\gamma=3$ (linear)	
		Served Customers	Unit Cost/ Customer	Served Customers	Unit Cost/ Customer	Served Customers	Unit Cost/ Customer	Served Customers	Unit Cost/ Customer
	3	160	53,4	150	63,8	150	63,9	150	66,3
	4	160	54,0	152	61,8	154	60,7	153	62,0
	5	156	55,4	144	65,7	145	65,4	147	63,8
	6	153	56,0	142	66,5	138	69,2	141	67,1
	7	154	56,4	140	67,2	139	69,3	142	68,2
L_rc101	1	98	78,3	98	80,5	93	87,2	97	83,2
	2	101	76,4	96	82,7	94	86,6	99	82,1
	3	100	76,7	96	82,2	94	86,0	99	81,5
	4	100	76,7	96	82,5	93	87,0	98	82,3
	5	103	75,1	97	81,6	96	84,0	97	83,0
	6	104	74,4	97	82,0	95	84,9	97	83,2
	7	102	76,0	99	79,7	95	84,9	97	83,2
L_rc105	1	104	73,6	102	78,1	100	80,6	104	77,2
	2	107	71,7	101	78,9	100	80,8	106	76,0
	3	106	72,4	102	77,6	100	80,5	105	76,8
	4	112	69,6	106	74,8	100	80,4	103	78,0
	5	113	69,4	104	76,5	100	80,7	105	77,0
	6	111	70,3	105	75,9	101	79,7	106	75,9
	7	111	70,6	103	77,6	102	79,2	107	75,3
L_rc107	1	110	69,2	105	75,8	101	79,9	107	74,9
	2	114	65,5	107	74,0	104	78,2	111	71,7
	3	117	65,3	107	73,8	105	76,5	111	71,8
	4	118	65,2	106	75,1	104	76,8	109	73,1
	5	116	65,8	112	70,6	104	76,8	109	73,3
	6	118	64,5	111	72,3	107	75,0	110	72,6
	7	119	64,6	107	74,9	106	75,7	111	71,5

---

# **THE MULTI-PERIOD VEHICLE ROUTING PROBLEM AND ITS APPLICATIONS**

**THEODORE ATHANASOPOULOS**

---