UNIVERSITY OF THE AEGEAN

SCHOOL OF BUSINESS

DEPARTMENT OF FINANCIAL AND MANAGEMENT ENGINEERING

« A Class of Single Vehicle Routing Problems with Predefined Customer Sequence and Depot Returns »

Ph.D Dissertation

Antonios Tatarakis

Supervisor: Ioannis Minis

December 2007

CHIOS

« A Class of Single Vehicle Routing Problems with Predefined Customer Sequence and Depot Returns »

Ph.D Dissertation

Antonios Tatarakis

Supervisor: Ioannis Minis

December 2007

CHIOS

Acknowledgements

During the time of research for this doctoral, I had the opportunity to cooperate with a number of people, each one of which assisted me in achieving this optimum goal. I therefore feel the need to acknowledge their contribution in the following lines.

I would like to start with my supervisor, Professor Ioannis Minis, for providing an unlimited source of information and advice, on both a scientific as well as a personal level, throughout the duration of my research time. His help, vision and solid scientific background, lead the way and catapulted this piece of research on a completely different level.

A special thank you to my colleagues Vassilios Zeimpekis from ELTRUN-WRC, Kostis Mammasis and Polybios Tsiribas from the Department of Financial Management Engineering at the University of the Aegean, for being there so that I could share my thoughts and for their advice and assistance at various phases of the project.

My final acknowledgement goes to my family, and especially to my wife, Mariza Balaska, for the continual moral support she provided throughout the duration of this work, for being the person she is, simply for being there.

Thank you

Εκτενής Περίληψη

Στην παρούσα διατριβή μελετήσαμε μια βασική περίπτωση του προβλήματος δρομολόγησης οχημάτων (Vehicle Routing Problem - VRP), στην οποία ένα όχημα ξεκινά από την αποθήκη και εξυπηρετεί πελάτες με *προκαθορισμένη σειρά επίσκεψης*, επιστρέφοντας στην αποθήκη για επαναφόρτωση όταν αυτό κρίνεται σκόπιμο. Στόχος είναι η εξυπηρέτηση όλων των πελατών και η ελαχιστοποίηση της διανυθείσας απόστασης (κόστους). Το συγκεκριμένο πρόβλημα έχει μεγάλο πρακτικό ενδιαφέρον, με ενδεικτικές εφαρμογές που περιλαμβάνουν την περίπτωση πωλήσεων Ex-Van, καθώς και συστήματα διακίνησης υλικών. Πέντε περιπτώσεις του παραπάνω προβλήματος, με αυξανόμενη πολυπλοκότητα, προτείνονται, αναλύονται και επιλύονται. Αυτές είναι:

- Διανομή πολλαπλών τύπων προϊόντων με γνωστή (deterministic) ζήτηση πελατών. Μελετήθηκαν δύο υπό-περιπτώσεις: α) κάθε τύπος προϊόντος φυλάσσεται σε ειδικό αποθηκευτικό χώρο οχήματος με προκαθορισμένη χωρητικότητα και β) όλοι οι τύποι προϊόντων αποθηκεύονται σε ένα (ενιαίο) χώρο.

- Διανομή πολλαπλών τύπων προϊόντων με στοχαστική (stochastic) ζήτηση πελατών. Μελετήθηκαν και οι δύο υποπεριπτώσεις που αναφέρονται παραπάνω. Σύμφωνα με αυτή την περίπτωση, η ζήτηση του κάθε πελάτη δεν είναι γνωστή εκ των προτέρων, αλλά αποκαλύπτεται μόλις το όχημα επισκεφτεί τον συγκεκριμένο πελάτη. Το συγκεκριμένο πρόβλημα είναι σημαντικά πιο σύνθετο. Στην περίπτωση που η ζήτηση του πελάτη δεν μπορεί να καλυφθεί πλήρως, το όχημα θα εξυπηρετήσει τον πελάτη μερικώς, θα επιστρέψει στην αποθήκη για επαναφόρτωση, και θα επανέλθει στον πελάτη ώστε να ικανοποιήσει και την εναπομένουσα ζήτησή του.

- Παραλαβή και διανομή (προϊόντων) με στοχαστική (stochastic) ζήτηση πελατών. Σε αυτή την περίπτωση το όχημα όχι μόνο παραδίδει προϊόντα στους πελάτες, αλλά και παραλαμβάνει επιστροφές από αυτούς (π.χ. κατεστραμμένα ή άδειες παλέτες ή υλικά συσκευασίας). Η ζήτηση του κάθε πελάτη για διανομή ή παραλαβή δεν είναι γνωστή εκ των προτέρων, αλλά αποκαλύπτεται μόλις το όχημα επισκεφτεί τον συγκεκριμένο πελάτη. Επιπρόσθετα, σε κάθε επιστροφή στην αποθήκη, θα πρέπει να αποφασισθεί πόσο απόθεμα θα φορτωθεί στο φορτηγό, ώστε να παραμείνει αρκετός άδειος χώρος για την παραλαβή των επιστρεφόμενων προϊόντων από τους επόμενους πελάτες.

Όπως αναφέρθηκε και προηγουμένως, οι πέντε παραπάνω περιπτώσεις παρουσιάζουν ιδιαίτερη πρακτική αξία στα Logistics (π.χ. πωλήσεις Ex-van) και σε συστήματα διακίνησης υλικών (material handling systems). Στην πρώτη περίπτωση (πωλήσεις Ex-van) ένα όχημα επισκέπτεται σε μια βάρδια έναν αριθμό πελατών, με προκαθορισμένη σειρά επίσκεψης και στοχαστική ζήτηση. Σκοπός του οχήματος είναι να εξυπηρετήσει πλήρως την ζήτηση όλων των πελατών, τηρώντας την σειρά επίσκεψης και επιστρέφοντας στην αποθήκη για επαναφόρτωση όποτε αυτό κρίνεται σκόπιμο. Η δεύτερη περίπτωση (material handling systems) βρίσκει εφαρμογή σε συστήματα παραγωγής με προκαθορισμένους διαδρόμους για αυτοκινούμενα οχήματα (automatic guided vehicles – AGVs). Η ζήτηση του κάθε σταθμού εργασίας μπορεί να είναι γνωστή εκ των προτέρων (συστήματα παραγωγής push – make to stock) ή στοχαστική (συστήματα παραγωγής pull – just in time). Και πάλι, σκοπός του οχήματος AGV είναι να εξυπηρετήσει πλήρως την ζήτηση όλων των σταθμών εργασίας, (τηρώντας την σειρά επίσκεψης) και επιστρέφοντας στην αποθήκη για επαναφόρτωση πρώτων υλών όποτε αυτό κρίνεται σκόπιμο.

Στο Κεφάλαιο 2 της διατριβής παρουσιάζουμε σημαντικά αποτελέσματα της βιβλιογραφίας που σχετίζονται με τα υπό διερεύνηση προβλήματα. Αρχικά αναλύεται το πρόβλημα δρομολόγησης οχημάτων (Vehicle Routing Problem - VRP). Συγκεκριμένα, παρουσιάζονται οι επεκτάσεις του προβλήματος αυτού με προκαθορισμένη χωρητικότητα (Capacitated VRP), με πολλαπλές παραδόσεις ανά πελάτη (Split Delivery VRP), με διανομή και παραλαβή (Pickup and Delivery VRP) και με στοχαστική ζήτηση (Stochastic VRP). Η δουλειά των Yang *et al.* (2000) αποτέλεσε έμπνευση για την παρούσα διατριβή. Οι Yang *et al.* διερεύνησαν το στοχαστικό VRP (SVRP) με ένα ή περισσότερα οχήματα, και προκαθορισμένη σειρά επίσκεψης. Σε αντίθεση με την συνηθισμένη πρακτική της βιβλιογραφίας σύμφωνα με την οποία όταν το όχημα δεν έχει πλέον αρκετό απόθεμα για να εξυπηρετήσει τους επόμενους πελάτες επιστρέφει στην αποθήκη για αναπλήρωση (recourse action) οι Yang *et al.* προτείνουν μια πολιτική βέλτιστης αναπλήρωσης αποθέματος η οποία ενσωματώνεται στην αρχική δρομολόγηση του οχήματος (proactive). Πιο συγκεκριμένα, τα σημεία αναπλήρωσης αποθέματος ενσωματώνονται σκοπίμως στην διαδρομή του οχήματος, ώστε η πιθανότητα αποτυχίας της διαδρομής, αλλά και το κόστος που αυτή η αποτυχία επιφέρει, να είναι μειωθεί, και το συνολικό αναμενόμενο κόστος της διαδρομής να ελαχιστοποιηθεί.

Τέλος, στο κεφάλαιο αυτό, εντοπίζουμε τα πεδία για περαιτέρω έρευνα στη συγκεκριμένη περιοχή, και ορίζουμε τα προβλήματα της παρούσας διατριβής, εξηγώντας την θεωρητική αλλά και πρακτική αξία αυτών.

Το Πρόβλημα Δρομολόγησης με Επιστροφές στην Αποθήκη (VRDRP)

Στο κεφάλαιο αυτό παρουσιάζουμε την βασική μορφή του Προβλήματος Δρομολόγησης με Επιστροφές στην Αποθήκη (VRDRP) με γνωστή εκ των προτέρων (ντετερμινιστική) ζήτηση. Σκοπός αυτού του προβλήματος είναι η ελαχιστοποίηση του κόστους (απόστασης) και η ταυτόχρονη εξυπηρέτηση όλων των πελατών με προκαθορισμένη σειρά επίσκεψης και ένα όχημα. Η αντικειμενική συνάρτηση του προβλήματος είναι η ακόλουθη:

$$\text{Min E} = \sum_{i=0}^{n-1} c_{i,i+1} x_{i,i+1} + \sum_{i=1}^{n-1} c_{i0} x_{i0} + \sum_{i=2}^{n} c_{0i} x_{0i} \qquad (\Pi\text{–}1)$$

όπου $c_{i,i+1}$ οι αποστάσεις μεταξύ των πελατών, c_{i0} μεταξύ πελατών και αποθήκης, και $x_{i,i+1}$, x_{i0} οι συντελεστές που ορίζουν αν ένα συγκεκριμένο τόξο είναι μέρος της διαδρομής. Επιπροσθέτως, εφαρμόζονται περιορισμοί δικτύου και χωρητικότητας του οχήματος ώστε το πρόβλημα να προσεγγίσει την πραγματικότητα. Η προτεινόμενη μέθοδος επίλυσης βασίζεται σε Δυναμικό Προγραμματισμό και ως πηγή έμπνευσης ήταν η δημοσίευση των Yang *et al.* (2000). Συγκεκριμένα, οι εξισώσεις δυναμικού προγραμματισμού που προτείνονται είναι οι ακόλουθες:

Για k= n:

$V_n(z) = x_{n,0}$ και $z = 0, 1\ldots Q\text{-}d_n$ (Π-2)

Για k= n-1:

$V_{n-1}(z) = x_{n-1,n} + V_n(z)$ αν $z \geq d_n$ (Π-3)

$\quad = x_{n-1,0} + x_{0,n} + V_n(z)$ αν $z < d_n$ (Π-4)

Για k= n-2…1:

$V_k(z) = x_{k,0} + x_{0,k+1} + V_{k+1}(Q\text{-}d_{k+1})$ αν $z < d_{k+1}$ (Π-5)

$\quad = \min\{ x_{k,0} + x_{0,k+1} + V_{k+1}(Q\text{-}d_{k+1}), x_{k,k+1} + V_{k+1}(z\text{-}d_{k+1})\}$ αν $z \geq d_{k+1}$ (Π-6)

όπου $V_k(z)$, k = n, n-1… 1 και z = 0… $Q\text{-}d_k$, η ελάχιστη απόσταση από τον πελάτη k, από τον οποίο το όχημα αναχωρεί με απόθεμα ίσο με z, μέχρι το τέλος της διαδρομής. Οι παραπάνω εξισώσεις λύνουν το πρόβλημα σταδιακά, ξεκινώντας από τον τελευταίο πελάτη, σύμφωνα με την μεθοδολογία του δυναμικού προγραμματισμού. Σε κάθε βήμα (από k= n-2…1), το πρόγραμμα υπολογίζει δύο τιμές, την τιμή αν το όχημα κατευθυνθεί απευθείας στον επόμενο πελάτη και αυτή εάν πάει στον επόμενο πελάτη μέσω της αποθήκης, και επιλέγει αυτή που παρέχει το ελάχιστο κόστος από το συγκεκριμένο βήμα μέχρι το τέλος της διαδρομής.

Η λεπτομερής ανάλυση του προβλήματος έδειξε ότι η πολυπλοκότητά του αυξάνει εκθετικά με το πλήθος των πελατών. Επιπροσθέτως, ο αριθμός των επιστροφών στην αποθήκη για αναπλήρωση αποθέματος στην βέλτιστη λύση g_{best} είναι τις περισσότερες φορές μεταξύ των $g_{min} \leq g_{best} \leq g_{min+2}$ όπου g_{min} είναι ο ελάχιστος εφικτός αριθμός επιστροφών στην αποθήκη. Ο αλγόριθμος δυναμικού προγραμματισμού αποδείχθηκε πολύ γρήγορος στην επίλυση μεγάλων προβλημάτων όπως ήταν αναμενόμενο. Για παράδειγμα, προβλήματα με 100 πελάτες επιλύθηκαν σχεδόν στιγμιαία, και προβλήματα με 1000 πελάτες επιλύθηκαν σε 0.3 δευτερόλεπτα, σε υπολογιστή με χαρακτηριστικά Intel Pentium IV, 1.6 GHz CPU, 1Gb RAM.

Παραλλαγές του Προβλήματος VRDRP

Στο κεφάλαιο 4 διερευνήσαμε: (i) την περίπτωση διανομής πολλαπλών τύπων προϊόντων στην οποία ο κάθε τύπος προϊόντος αποθηκεύεται σε διαφορετικό αποθηκευτικό χώρο στο όχημα, και (ii) την περίπτωση διανομής πολλαπλών τύπων προϊόντων στην οποία όλα τα προϊόντα αποθηκεύονται σε έναν αποθηκευτικό χώρο.

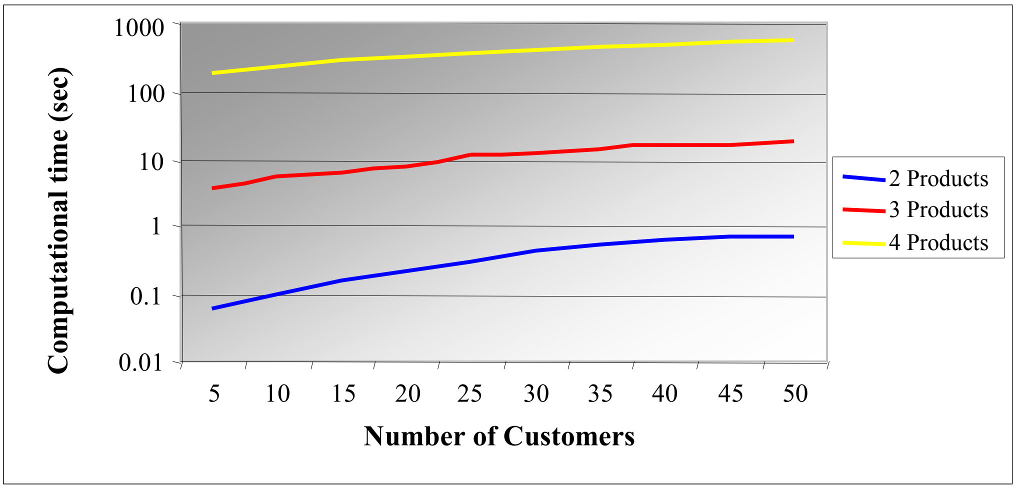
Και τα δύο προβλήματα επιλύθηκαν με κατάλληλες επεκτάσεις του αλγορίθμου δυναμικού προγραμματισμού που παρουσιάστηκε παραπάνω. Η εξίσωση δυναμικού προγραμματισμού για την πρώτη περίπτωση έχει ως εξής:

$$V_i(z_1,\ldots,z_K) = \begin{cases} c_{i0} + c_{0,i+1} + V_{i+1}(Q_1 - d_{1,i+1},\ldots,Q_K - d_{K,i+1}), & \text{if } \exists\, j \in \{1,\ldots,K\} : z_j < d_{j,i+1}, \\ \\ \min\{c_{i0} + c_{0,i+1} + V_{i+1}(Q_1 - d_{1,i+1},\ldots,Q_K - d_{K,i+1}), & \\ c_{i,i+1} + V_{i+1}(z_1 - d_{1,i+1},\ldots,z_K - d_{K,i+1})\}, & \text{if } \forall\, j \in \{1,\ldots,K\} : z_j \geq d_{j,i+1}. \end{cases} \tag{Π-7}$$

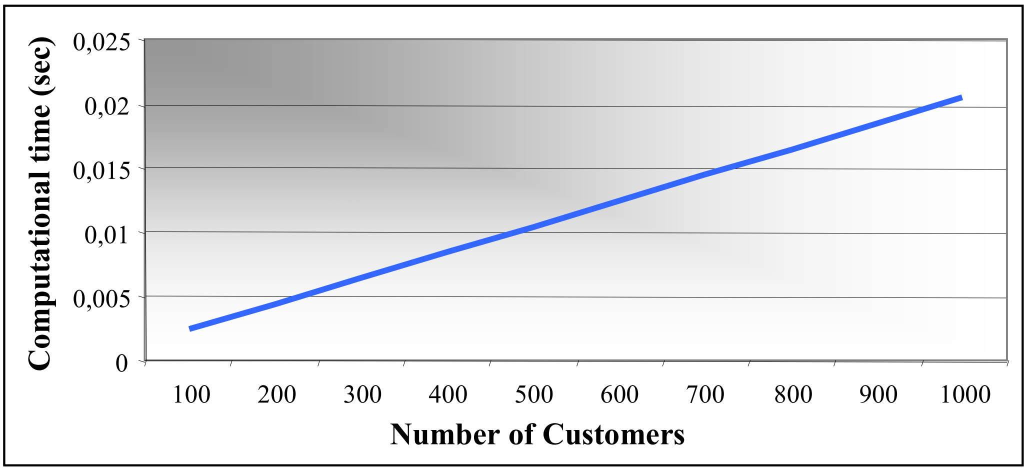
όπου $d_{j,i}$ *(j=1,…K)* η ζήτηση για το προϊόν *j* από τον πελάτη *i*, $V_i(z_1,\ldots,z_K), i=1,\ldots,n,$ το ελάχιστο κόστος από τον πελάτη *i* μέχρι το τέλος της διαδρομής, εάν ο πελάτης *i* έχει εξυπηρετηθεί και το απόθεμα που έχει απομείνει στο όχημα για το προϊόν $j, 1 \le j \le K,$ είναι $z_j \in \{0,\ldots,Q_j - d_{ji}\}$.

Η δεύτερη περίπτωση κατά την οποία όλα τα προϊόντα αποθηκεύονται σε έναν αποθηκευτικό χώρο επιλύθηκε και αυτή με δυναμικό προγραμματισμό. Το πρόβλημα αρχικά μετασχηματίστηκε στο αντίστοιχο VRRDP, αθροίζοντας την ζήτηση για το κάθε προϊόν σε κάθε πελάτη ώστε η συνολική ζήτηση του πελάτη να αναπαριστάται από ένα νούμερο. Για την επίλυση του προβλήματος χρησιμοποιήθηκε ο αλγόριθμος που παρουσιάστηκε στην Ενότητα 3.4 αυτής της διατριβής, ο οποίος σχεδιάστηκε για την επίλυση του VRRDP με ένα προϊόν. Ο αλγόριθμος αυτός υπολογίζει την διαδρομή του οχήματος και με μια μετατροπή σε αυτόν υπολογίζονται και οι βέλτιστες ποσότητες που πρέπει να φορτωθούν στο όχημα, σε κάθε του επιστροφή στην αποθήκη. Αυτό γίνεται με τον συνδυασμό της γνώσης της επιμέρους ζήτησης ανά προϊόν ανά πελάτη (από τα αρχικά δεδομένα του προβλήματος), και της βέλτιστης διαδρομής που θα ακολουθήσει το όχημα (που έχει υπολογιστεί από τον αλγόριθμο). Γνωρίζοντας δυο διαδοχικές επιστροφές στην αποθήκη μέσα στην διαδρομή, μπορεί να υπολογιστεί το φορτίο που πρέπει να μεταφέρει το όχημα για να ικανοποιήσει πλήρως την ζήτηση του κάθε πελάτη για κάθε προϊόν.

Το κεφάλαιο ολοκληρώνεται με την ανάλυση της απόδοσης των προτεινόμενων μεθόδων λύσης. Συγκεκριμένα επιλύθηκαν 3000 προβλήματα για την πρώτη και 2000 προβλήματα για την δεύτερη περίπτωση.



Σχήμα Π.1. Οι υπολογιστικοί χρόνοι της πρώτης περίπτωσης.



Σχήμα Π.2. Οι υπολογιστικοί χρόνοι της δεύτερης περίπτωσης.

Βάσει των αποτελεσμάτων αυτών, δείξαμε ότι η πολυπλοκότητα του προβλήματος διανομής πολλαπλών προϊόντων στην οποία ο κάθε τύπος προϊόντος αποθηκεύεται σε διαφορετικό αποθηκευτικό χώρο στο όχημα είναι σημαντικά μεγαλύτερη από αυτή του προβλήματος διανομής πολλαπλών προϊόντων στην οποία όλοι οι τύποι προϊόντος αποθηκεύονται σε έναν αποθηκευτικό χώρο (χύδην φορτίο).

Το Στοχαστικό Πρόβλημα Δρομολόγησης με Επιστροφές στην Αποθήκη (SVRDRP)

Στο Κεφάλαιο 5 αναλύουμε την στοχαστική έκδοση του Προβλήματος Δρομολόγησης με Επιστροφές στην Αποθήκη (SVRDRP). Σε αυτό το πρόβλημα η ζήτηση των πελατών μοντελοποιείται ως ανεξάρτητη τυχαία μεταβλητή με γνωστές στατιστικές παραμέτρους (βάσει ιστορικής ζήτησης). Σκοπός του κεφαλαίου είναι η ανάλυση του προβλήματος όσο αφορά στις στατιστικές παραμέτρους της ζήτησης. Η εξίσωση δυναμικού προγραμματισμού για το στοχαστικό VRDRP σύμφωνα με τους Yang *et al.* (2000) δίδεται παρακάτω:

$$f_j(z) = \min \begin{cases} c_{j,j+1} + \sum_{k:\xi^k \le z} f_{j+1}(z-\xi^k)p_{j+1,k} + \sum_{k:\xi^k > z} [2c_{j+1,0} + f_{j+1}(z+Q-\xi^k)]p_{j+1,k} \\ \\ c_{j,0} + c_{0,j+1} + \sum_{k=1}^{m} f_{j+1}(Q-\xi^k)p_{j+1,k} \end{cases} \qquad (\Pi\text{-}8)$$

όπου $f_j(z)$ το ελάχιστο κόστος από τον πελάτη j μέχρι το τέλος της διαδρομής, εάν το απόθεμα που έχει απομείνει στο όχημα μετά την εξυπηρέτηση του πελάτη j είναι z και η στοχαστική ζήτηση του πελάτη $j+1$ είναι ξ^k. Επιπρόσθετα, Q η χωρητικότητα του οχήματος και $p_{j+1,k}$ η πιθανότητα να έχει ο πελάτης $j+1$ την ζήτηση ξ^k.

Αρχικά, το πρόβλημα αναλύθηκε ώστε να εντοπίσουμε την επίδραση της διακύμανσης της ζήτησης των πελατών στο αναμενόμενο ελάχιστο κόστος της διαδρομής. Η ανάλυση καταδεικνύει ότι το αναμενόμενο ελάχιστο κόστος της διαδρομής αυξάνει σχεδόν γραμμικά με την διακύμανση της ζήτησης. Έτσι, στην περίπτωση του Ex-van, η συνέπεια των πωλήσεων επηρεάζει άμεσα το κόστος διανομής. Επιπροσθέτως το πρόβλημα αναλύθηκε με σκοπό να καθοριστεί η σχέση μεταξύ του μέσου όρου της ζήτησης των πελατών και της διακύμανσης της ζήτησης. Στο συγκεκριμένο παράδειγμα που παρουσιάζεται, η ποσοστιαία αύξηση του ελάχιστου αναμενόμενου κόστους της διαδρομής (11.6%) για την περίπτωση με τον χαμηλό μέσο όρο ζήτησης είναι μικρότερη από την αντίστοιχη ποσοστιαία αύξηση (16.7%) για την περίπτωση με τον υψηλό μέσο όρο. Συμπερασματικά, η τυχαιότητα επηρεάζει το αναμενόμενο ελάχιστο κόστος της διαδρομής περισσότερο σε οχήματα μικρής χωρητικότητας.

Επεκτάσεις του προβλήματος SVRDRP

Στο Κεφάλαιο 6 επεκτείνουμε το Στοχαστικό Πρόβλημα Δρομολόγησης με Επιστροφές στην Αποθήκη (SVRDRP) για να επιλύσουμε την περίπτωση διανομής πολλαπλών προϊόντων. Όπως και στο Κεφάλαιο 4 διερευνήθηκαν δύο περιπτώσεις: (i) η περίπτωση διανομής πολλαπλών τύπων προϊόντων στην οποία ο κάθε τύπος προϊόντος αποθηκεύεται σε διαφορετικό αποθηκευτικό χώρο στο όχημα, και (ii) η περίπτωση διανομής πολλαπλών τύπων προϊόντων στην οποία όλα τα προϊόντα αποθηκεύονται σε ενιαίο αποθηκευτικό χώρο. Στο κεφάλαιο αυτό παρουσιάζουμε τα χαρακτηριστικά του κάθε προβλήματος, νέες μεθόδους για τον υπολογισμό του ελάχιστου αναμενόμενου κόστους, και θεωρητικά αποτελέσματα τα οποία μας επιτρέπουν τον προσδιορισμό της βέλτιστης απόφασης δρομολόγησης μετά την εξυπηρέτηση του κάθε πελάτη.

Όσον αφορά την πρώτη περίπτωση, η εξίσωση δυναμικού προγραμματισμού για δύο προϊόντα είναι η ακόλουθη:

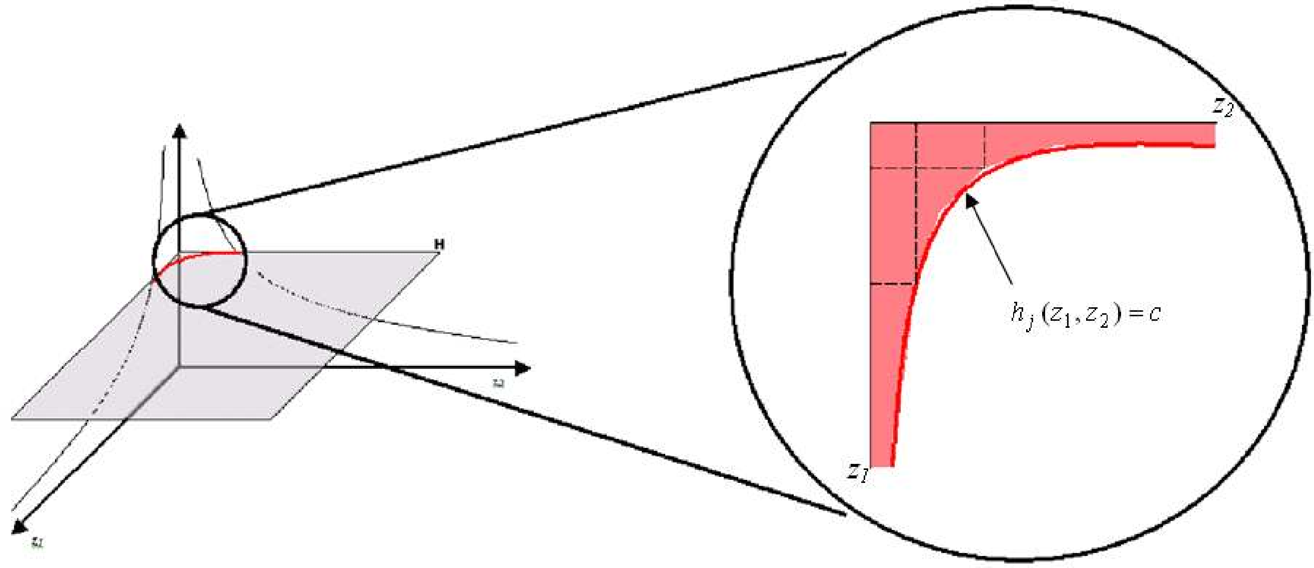
$$f_j(z_1,z_2)=\min\begin{cases} c_{j,j+1}+\sum\limits_{k_1:\xi^{k_1}\le z_1}\sum\limits_{k_2:\xi^{k_2}\le z_2} f_{j+1}(z_1-\xi^{k_1},z_2-\xi^{k_2})p^1_{j+1,k_1}p^2_{j+1,k_2}+ \\ +\sum\limits_{k_1:\xi^{k_1}>z_1}\sum\limits_{k_2:\xi^{k_2}\le z_2}\left[2c_{j+1,0}+f_{j+1}(z_1+Q_1-\xi^{k_1},Q_2)\right]p^1_{j+1,k_1}p^2_{j+1,k_2}+ \\ +\sum\limits_{k_1:\xi^{k_1}\le z_1}\sum\limits_{k_2:\xi^{k_2}>z_2}\left[2c_{j+1,0}+f_{j+1}(Q_1,z_2+Q_2-\xi^{k_2})\right]p^1_{j+1,k_1}p^2_{j+1,k_2}+ \\ +\sum\limits_{k_1:\xi^{k_1}>z_1}\sum\limits_{k_2:\xi^{k_2}>z_2}\left[2c_{j+1,0}+f_{j+1}(z_1+Q_1-\xi^{k_1},z_2+Q_2-\xi^{k_2})\right]p^1_{j+1,k_1}p^2_{j+1,k_2} \\ \\ c_{j,0}+c_{0,j+1}+\sum\limits_{k_1=1}^{m_1}\sum\limits_{k_2=1}^{m_2} f_{j+1}(Q_1-\xi^{k_1},Q_2-\xi^{k_2})p^1_{j+1,k_1}p^2_{j+1,k_2} \end{cases} \quad (Π\text{-}9)$$

όπου $f_j(z_1, z_2)$ το ελάχιστο κόστος από τον πελάτη j μέχρι το τέλος της διαδρομής, εάν το απόθεμα που έχει απομείνει στο όχημα είναι (z_1,z_2) και η στοχαστική ζήτηση του πελάτη $j+1$ είναι (ξ^{k_1},ξ^{k_2}). Επιπρόσθετα, Q_1 και Q_2 η χωρητικότητα του κάθε αποθηκευτικού χώρου του οχήματος, και $(p^1_{j+1,k_1},p^2_{j+1,k_2})$ οι πιθανότητες να έχει ο πελάτης $j+1$ την αντίστοιχη ζήτηση (ξ^{k_1},ξ^{k_2}).

Επιπλέον αποδείχθηκε ότι για κάθε πελάτη υπάρχει μια συνάρτηση κρίσιμων σημείων, η οποία διαχωρίζει δυο περιοχές στον χώρο πιθανών φορτίων μετά την εξυπηρέτηση του πελάτη: Την περιοχή συνδυασμών φορτίων για τους οποίους η βέλτιστη απόφαση (μετά την εξυπηρέτηση του πελάτη) είναι να επιστρέψει το όχημα στην αποθήκη, και την περιοχή για την οποία η βέλτιστη απόφαση είναι το όχημα να συνεχίσει στον επόμενο πελάτη. Το αποτέλεσμα αυτό βασίζεται στο ακόλουθο θεώρημα:

ΘΕΩΡΗΜΑ 1: *Για κάθε πελάτη j, υπάρχει μια συνάρτηση κρίσιμων σημείων* $h_j(z_1^*,z_2^*)=c_j$*, τέτοια ώστε η βέλτιστη απόφαση, μετά την πλήρη εξυπηρέτηση του πελάτη j είναι να προχωρήσει το όχημα στον επόμενο πελάτη j+1 εάν* $h_j(z_1,z_2)\ge c_j$ *αλλιώς να επιστρέψει στην αποθήκη.*

Η συνάρτηση $h_j(z_1^*,z_2^*)=c_j$ αποτυπώνεται γραφικά στο παρακάτω Σχήμα Π-3.



Σχήμα Π-3. Γραφική αναπαράσταση της συνάρτησης ορίου.

Η απόδειξη του παραπάνω θεωρήματος στηρίζεται στο γεγονός ότι το πρώτο μέρος της εξίσωσης δυναμικού προγραμματισμού είναι μονότονα μειούμενη και το δεύτερο μέρος της εξίσωσης είναι σταθερά, ανεξάρτητη του φορτίου (z_1, z_2). Έτσι οι δύο όροι της εξίσωσης τέμνονται σε μια γραμμή που ορίζει την συνάρτηση $h_j(z_1^*, z_2^*) = c_j$, όπως φαίνεται στο Σχήμα Π-3.

Για την απόδειξη της μονοτονικότητας του πρώτου όρου της Εξίσωσης (Π-9), χρησιμοποιείται το ακόλουθο λήμμα, το οποίο αναπτύσσεται και αποδεικνύεται στην Ενότητα 6.2.2.:

ΛΗΜΜΑ 1: $f_j(z_1, z_2) \leq f_j(Q_1, Q_2) + 2c_{0j}$ *για κάθε* $z_1, z_2 \in S_j$

Όσον αφορά την δεύτερη περίπτωση, στην οποία όλοι οι τύποι προϊόντος αποθηκεύονται σε έναν αποθηκευτικό χώρο η εξίσωση δυναμικού προγραμματισμού είναι η ακόλουθη::

$f_j(z_1, z_2) = \min$

$$
\left\{
\begin{array}{l}
c_{j,j+1} + \sum\limits_{k_1:\xi^{k_1}\le z_1} \sum\limits_{k_2:\xi^{k_2}\le z_2} f_{j+1}(z_1-\xi^{k_1}, z_2-\xi^{k_2}) p_{j+1}(k_1,k_2) + \\
+ \sum\limits_{k_1:z_1<\xi^{k_1}} \sum\limits_{k_2:\xi^{k_2}\le z_2} \left[2c_{j+1,0} + \min\limits_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}(\theta-(\xi^{k_1}-z_1), Q-\theta) \right] p_{j+1}(k_1,k_2) + \\
+ \sum\limits_{k_1:\xi^{k_1}\le z_1} \sum\limits_{k_2:z_2<\xi^{k_2}} \left[2c_{j+1,0} + \min\limits_{\xi^{k_2}-z_2\le\theta\le Q} f_{j+1}(\theta, Q-\theta-(\xi^{k_2}-z_2)) \right] p_{j+1}(k_1,k_2) + \\
+ \sum\limits_{k_1:z_1<\xi^{k_1}} \sum\limits_{k_2:z_2<\xi^{k_2}} \left[2c_{j+1,0} + \min\limits_{\xi^{k_1}-z_1\le\theta\le Q-(\xi^{k_2}-z_2)} f_{j+1}(\theta-(\xi^{k_1}-z_1), Q-\theta-(\xi^{k_2}-z_2)) \right] p_{j+1}(k_1,k_2) \\
\\
\min\limits_{0\le\theta\le Q} \left\{
\begin{array}{l}
c_{j,0} + c_{0,j+1} + \\
\sum\limits_{\xi^{k_1}\le\theta} \sum\limits_{\xi^{k_2}\le Q-\theta} f_{j+1}(\theta-\xi^{k_1}, Q-\theta-\xi^{k_2}) p_{j+1}(k_1,k_2) + \\
+ \sum\limits_{\theta<\xi^{k_1}} \sum\limits_{\xi^{k_2}\le Q-\theta} \left[2c_{j+1,0} + \min\limits_{\xi^{k_1}-\theta\le s\le Q} f_{j+1}(s-(\xi^{k_1}-\theta), Q-s) \right] p_{j+1}(k_1,k_2) + \\
+ \sum\limits_{\xi^{k_1}\le\theta} \sum\limits_{Q-\theta<\xi^{k_2}} \left[2c_{j+1,0} + \min\limits_{0\le s\le 2Q-\theta-\xi^{k_2}} f_{j+1}\{s, Q-s-[\xi^{k_2}-(Q-\theta)]\} \right] p_{j+1}(k_1,k_2)
\end{array}
\right\}
\end{array}
\right. \tag{Π-10}
$$

όπου $f_j(z_1, z_2)$ το ελάχιστο κόστος από τον πελάτη j μέχρι το τέλος της διαδρομής, εάν το απόθεμα που έχει απομείνει στο όχημα είναι (z_1, z_2) και η στοχαστική ζήτηση του πελάτη $j+1$ είναι (ξ^{k_1}, ξ^{k_2}). Επιπρόσθετα, Q η χωρητικότητα του οχήματος, και $p_{j+1}(k_1, k_2)$ η πιθανότητα να έχει ο πελάτης $j+1$ την ζήτηση (ξ^{k_1}, ξ^{k_2}).

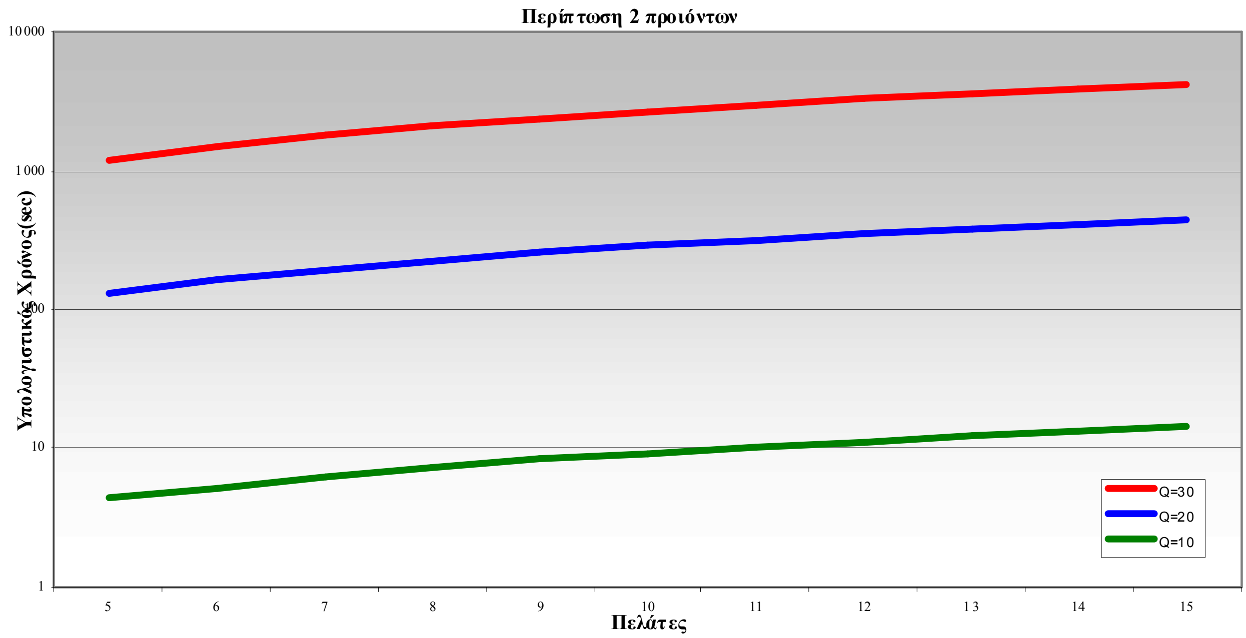
Αντίστοιχα, για την περίπτωση διανομής πολλαπλών προϊόντων στην οποία όλοι οι τύποι προϊόντος αποθηκεύονται σε έναν αποθηκευτικό χώρο (χύδην φορτίο) αναπτύχθηκε και αποδείχθηκε το συγγενές με το Θεώρημα 1, Θεώρημα 2:

ΘΕΩΡΗΜΑ 2: *Για κάθε πελάτη j, υπάρχει μια συνάρτηση ορίου* $h^u{}_j(z_1, z_2) = c^u{}_j$, *τέτοια ώστε η βέλτιστη απόφαση, μετά την πλήρη εξυπηρέτηση του πελάτη j είναι να προχωρήσει το όχημα στον επόμενο πελάτη j+1 εάν* $h^u{}_j(z_1{}^*, z_2{}^*) \ge c^u{}_j$, *αλλιώς να επιστρέψει στην αποθήκη.*

Με βάση το δεύτερο αυτό θεώρημα, μπορεί να προσδιοριστεί η βέλτιστη απόφαση για τον προορισμό του οχήματος αφότου εξυπηρετήσει τον πελάτη *j*. Εάν ο συνδυασμός των φορτίων (z_1, z_2) είναι τέτοιος ώστε $h^u{}_j(z_1{}^*, z_2{}^*) \ge c^u{}_j$ τότε το όχημα πρέπει να προχωρήσει στον επόμενο πελάτη. Εάν δεν ικανοποιείται η παραπάνω ανισότητα, τότε το όχημα πρέπει να επιστρέψει στην αποθήκη. Και σε αυτή την περίπτωση η συνάρτηση $h^u{}_j(z_1, z_2) = c^u{}_j$ είναι η τομή των δύο όρων της Εξίσωσης (Π-10).

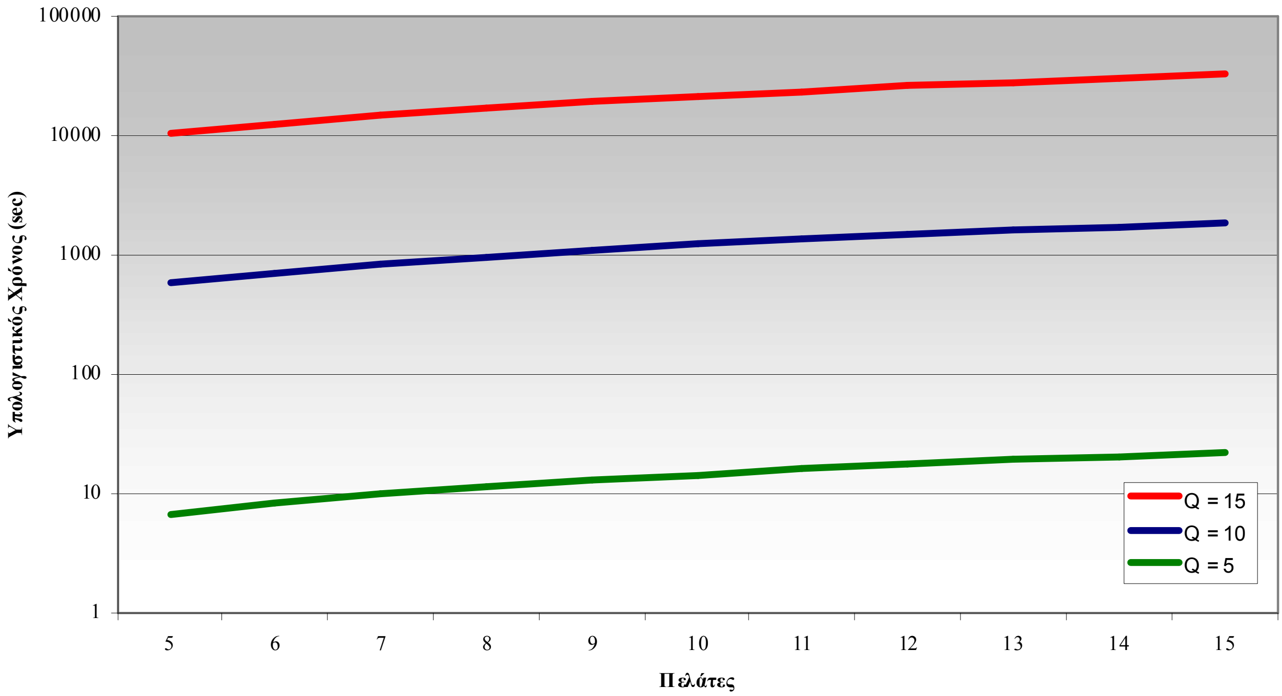
Και οι δύο περιπτώσεις μοντελοποιήθηκαν και επιλύθηκαν για δύο προϊόντα, αλλά τα αποτελέσματα μπορούν να επεκταθούν σε *n* προϊόντα (βλέπε Appendix B για την μαθηματική διατύπωση του προβλήματος με 3 προϊόντα στην οποία ο κάθε τύπος προϊόντος αποθηκεύεται σε διαφορετικό αποθηκευτικό χώρο στο όχημα).

Η απόδοση των προτεινόμενων μεθόδων αναλύθηκε με την επίλυση σημαντικού αριθμού προβλημάτων ανά περίπτωση (30,000 ανά περίπτωση). Και για τις δυο περιπτώσεις, μέσω την εκτέλεσης σημαντικού αριθμού τυχαίως δημιουργημένων προβλημάτων, βρέθηκε ότι η αύξηση της χωρητικότητας του οχήματος έχει ως αποτέλεσμα την σχεδόν εκθετική αύξηση του υπολογιστικού χρόνου της λύσης. Από την άλλη, εάν η χωρητικότητα του οχήματος παραμείνει σταθερή, ο υπολογιστικός χρόνος του αλγορίθμου αυξάνεται σχεδόν γραμμικά με το πλήθος των πελατών. Τα αποτελέσματα του αλγορίθμου της περίπτωσης διανομής δυο προϊόντων στην οποία ο κάθε τύπος προϊόντος αποθηκεύεται σε διαφορετικό αποθηκευτικό χώρο στο όχημα (συνολική χωρητικότητα του οχήματος *Q*) φαίνονται στο Σχήμα Π-4.



Σχήμα Π-4. Τα αποτελέσματα του αλγορίθμου για 2 προϊόντα που αποθηκεύονται ξεχωριστά.

Η δεύτερη περίπτωση αποθήκευσης προϊόντων σε ένα ενιαίο χώρο αποδείχθηκε σημαντικά πιο σύνθετη. Αυτό οφείλεται στα επιπρόσθετα βήματα ελαχιστοποίησης του μοντέλου γραμμικού προγραμματισμού, ώστε να εντοπισθούν οι βέλτιστες ποσότητες αποθέματος που πρέπει να φορτωθούν στο όχημα, μετά από κάθε επιστροφή αυτού στην αποθήκη. Η διερεύνηση της απόδοσης του αλγορίθμου της περίπτωσης αυτής παρουσιάζονται στο Σχήμα Π-5. Είναι ξεκάθαρο ότι η αύξηση της χωρητικότητας του οχήματος έχει ως αποτέλεσμα την σχεδόν εκθετική αύξηση του υπολογιστικού χρόνου της λύσης. Από την άλλη, εάν η χωρητικότητα του οχήματος παραμείνει σταθερή, ο υπολογιστικός χρόνος του αλγορίθμου αυξάνεται σχεδόν γραμμικά με το πλήθος των πελατών.



Σχήμα Π-5. Τα αποτελέσματα του αλγορίθμου για χύδην φορτίο.

Το VRDRP με Στοχαστικές Διανομές και Παραλαβές

Στο Κεφάλαιο 7 εξετάζουμε την περίπτωση Διανομών και Παραλαβών του VRDRP με άγνωστη εκ των προτέρων (στοχαστική) ζήτηση. Σε αυτή την περίπτωση το όχημα δεν διανέμει μόνο προϊόντα αλλά και παραλαμβάνει επιστρεφόμενα προϊόντα από τους πελάτες που επισκέπτεται (π.χ. άδειες παλέτες, προϊόντα που έχουν λήξει ή που έχουν καταστραφεί). Σκοπός του προβλήματος είναι η ελαχιστοποίηση της διανυθείσας απόστασης (κόστους) υπό στοχαστική ζήτηση τόσο για τα προϊόντα διανομής όσο και για αυτά της παραλαβής.

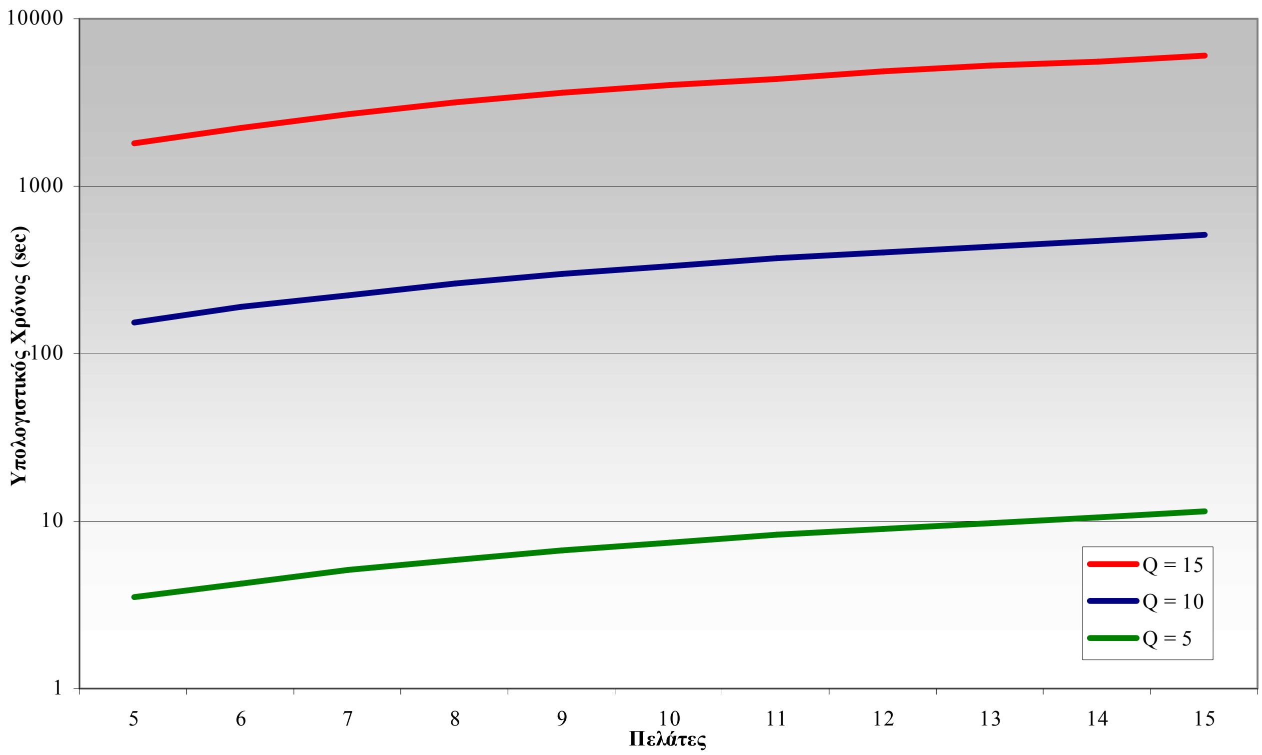
Σε αυτή την περίπτωση πρέπει να ληφθούν υπόψη επιπλέον παράγοντες. Μετά την επίσκεψη του οχήματος στην αποθήκη, πρέπει να ληφθεί μια επιπλέον απόφαση, σχετικά με την ποσότητα του προϊόντος για πώληση που θα φορτωθεί στο όχημα. Αυτό συμβαίνει λόγω του γεγονότος ότι το όχημα δεν μπορεί απλά να φορτωθεί έως το μέγιστο της χωρητικότητάς του, καθότι πρέπει να προβλεφθεί κενός χώρος ώστε το όχημα να μπορεί να παραλάβει και επιστρεφόμενα προϊόντα, αποφεύγοντας επιπρόσθετες επιστροφές στην αποθήκη.

Η συνάρτηση δυναμικού προγραμματισμού για την περίπτωση Διανομών και Παραλαβών του VRDRP με άγνωστη εκ των προτέρων (στοχαστική) ζήτηση δίδεται παρακάτω:

$$
f_j(z,b) = \min \left\{ \begin{array}{l}
c_{j,j+1} + \sum_{k:\xi^k \le z} \sum_{m:\rho^m + b \le Q-(z-\xi^k)} f_{j+1}(z-\xi^k, b+\rho^m) p_{j+1,k}\pi_{j+1,m} \\
+ \sum_{k:z<\xi^k} \sum_{m:\rho^m + b \le Q} \left[2c_{j+1,0} + \min_{\xi^k - z \le \theta \le Q} f_{j+1}(\theta - (\xi^k - z), 0) \right] p_{j+1,k}\pi_{j+1,m} \\
+ \sum_{k:\xi^k \le z} \sum_{m:Q-(z-\xi^k)<\rho^m + b} \left[2c_{j+1,0} + \min_{0\le\theta\le Q-[\rho^m - Q + (z-\xi^k)+b]} f_{j+1}[\theta, \rho^m - [Q-(z-\xi^k)-b]] \right] p_{j+1,k}\pi_{j+1,m} \\
+ \sum_{k:z<\xi^k} \sum_{m:Q<\rho^m + b} \left[2c_{j+1,0} + \min_{\xi^k - z \le \theta \le 2Q - z + \xi^k - \rho^m - b} f_{j+1}[\theta - (\xi^k - z), [\rho^m - (Q-b)]] \right] p_{j+1,k}\pi_{j+1,m} \\
\\
\min_{0\le\theta\le Q} \left\{ \begin{array}{l}
c_{j,0} + c_{0,j+1} \\
+ \sum_{k:\xi^k \le \theta} \sum_{m:\rho^m + \theta - \xi^k \le Q} f_{j+1}(\theta - \xi^k, \rho^m) p_{j+1,k}\pi_{j+1,m} \\
+ \sum_{k:\theta<\xi^k} \sum_{m:\rho^m \le Q} \left[2c_{j+1,0} + \min_{\xi^k - \theta \le s \le Q} f_{j+1}(s - (\xi^k - \theta), 0) \right] p_{j+1,k}\pi_{j+1,m} \\
+ \sum_{k:\xi^k \le \theta} \sum_{m:Q-(\theta-\xi^k)<\rho^m} \left[2c_{j+1,0} + \min_{0\le s\le Q-[\rho^m - Q + (\theta-\xi^k)]} f_{j+1}[s, \rho^m - [Q-(\theta-\xi^k)]] \right] p_{j+1,k}\pi_{j+1,m}
\end{array} \right\}
\end{array} \right. \qquad (\Pi\text{-}11)
$$

όπου $f_j(z, b)$ το ελάχιστο κόστος από τον πελάτη *j* μέχρι το τέλος της διαδρομής, εάν το απόθεμα που έχει απομείνει στο όχημα είναι *z* για το προϊόν διανομής και *b* για το προϊόν παραλαβής και η στοχαστική ζήτηση του πελάτη *j+1* είναι (ξ^k, ρ^m) αντίστοιχα. Επιπρόσθετα, Q η χωρητικότητα του οχήματος, και $(p_{j+1,k}, \pi_{j+1,m})$ οι πιθανότητες να έχει ο πελάτης *j+1* την αντίστοιχη ζήτηση (ξ^k, ρ^m).

Στην περίπτωση που η επιστροφή στην αποθήκη γίνει μετά την εξυπηρέτηση του πελάτη *j+1* τότε η ποσότητα του προϊόντος που θα φορτωθεί στο όχημα μπορεί να είναι τέτοια ώστε η ζήτηση του πελάτη *j+1* να ικανοποιηθεί πλήρως (για προϊόντα παράδοσης ή παραλαβής). Εάν όμως, η επιστροφή στην αποθήκη συμβεί πριν την εξυπηρέτηση του πελάτη *j+1* τότε μια επιπλέον (αμέσως επόμενη) επιστροφή στην αποθήκη θα καταστεί απαραίτητη στην περίπτωση που η ποσότητα που φορτώθηκε στο όχημα (ή ο χώρος που είχε μείνει για τα προϊόντα που θα παραληφθούν) δεν είναι επαρκής να ικανοποιήσει πλήρως την ζήτηση του συγκεκριμένου πελάτη.



Σχήμα Π-6. Τα αποτελέσματα του αλγορίθμου της περίπτωσης διανομών και παραλαβών.

Δια της εκτέλεσης σημαντικού αριθμού (30,000) τυχαίως δημιουργημένων προβλημάτων για την περίπτωση των διανομών και παραλαβών του SVRDRP (Σχήμα Π-6), βρέθηκε ότι η αύξηση της χωρητικότητας του οχήματος έχει ως αποτέλεσμα την σχεδόν εκθετική αύξηση του υπολογιστικού χρόνου του αλγορίθμου. Από την άλλη, εάν η χωρητικότητα του οχήματος παραμείνει σταθερή, ο υπολογιστικός χρόνος του αλγορίθμου αυξάνεται σχεδόν γραμμικά με το πλήθος των πελατών.

Συμπεράσματα

Στην παρούσα διατριβή μελετήθηκε μια βασική περίπτωση του προβλήματος δρομολόγησης οχημάτων (Vehicle Routing Problem - VRP), στην οποία ένα όχημα ξεκινά από την αποθήκη και εξυπηρετεί πελάτες με *προκαθορισμένη σειρά επίσκεψης*, επιστρέφοντας στην αποθήκη για επαναφόρτωση όταν αυτό κρίνεται σκόπιμο. Στόχος είναι η εξυπηρέτηση όλων των πελατών και η ελαχιστοποίηση της διανυθείσας απόστασης (κόστους). Η συνεισφορά της διατριβής συνοψίζεται στα ακόλουθα:

- Για πρώτη φορά μελετήθηκε το πρόβλημα διανομής πολλαπλών προϊόντων με προκαθορισμένη σειρά επίσκεψης, και οι δύο υπό-περιπτώσεις αυτού: α) κάθε τύπος προϊόντος να φυλάσσεται σε ειδικό αποθηκευτικό χώρο οχήματος με προκαθορισμένη χωρητικότητα και β) όλοι οι τύποι προϊόντων να αποθηκεύονται στον ένα (ενιαίο) χώρο του οχήματος. Προτάθηκε αλγόριθμος βέλτιστης επίλυσής του.

- Για πρώτη φορά αναλύθηκε το πρόβλημα διανομής προϊόντος με προκαθορισμένη σειρά επίσκεψης και στοχαστική ζήτηση, όσο αφορά στην επίδραση της διακύμανσης αλλά και του μέσου όρου της ζήτησης στο ελάχιστο αναμενόμενο κόστος της διαδρομής.

- Για πρώτη φορά μελετήθηκε το πρόβλημα διανομής πολλαπλών προϊόντων με προκαθορισμένη σειρά επίσκεψης με στοχαστική ζήτηση, και οι δύο υπό-περιπτώσεις αυτού: α) κάθε τύπος προϊόντος να φυλάσσεται σε ειδικό αποθηκευτικό χώρο οχήματος με προκαθορισμένη χωρητικότητα και β) όλοι οι τύποι προϊόντων να αποθηκεύονται στον ένα (ενιαίο) χώρο του οχήματος. Προτάθηκε αλγόριθμος ανεύρεσης του ελάχιστου αναμενόμενου κόστους της διαδρομής και πολιτική ανεύρεσης της διαδρομής η οποία στηρίχτηκε στα Θεωρήματα που αναπτύχθηκαν και παρουσιάζονται στην διατριβή.

- Για πρώτη φορά μελετήθηκε το πρόβλημα διανομής και παραλαβής προϊόντων με προκαθορισμένη σειρά επίσκεψης και στοχαστική ζήτηση. Προτάθηκε αλγόριθμος ανεύρεσης του ελάχιστου αναμενόμενου κόστους της διαδρομής και πολιτική ανεύρεσης της διαδρομής.

Τα αποτελέσματα της παρούσας διατριβής μπορούν να χρησιμοποιηθούν σε ένα σύστημα υποστήριξης λήψης αποφάσεων, για μια πληθώρα περιπτώσεων (γνωστή ή άγνωστη ζήτηση πελατών, ένα ή πολλαπλά προϊόντα, παραδώσεις, ή παραδώσεις και παραλαβές): Με αυτό τον τρόπο μπορούν να εξαλειφθούν οι τυχαίες αποφάσεις δρομολόγησης, ελαχιστοποιώντας τα συνολικά λειτουργικά κόστη της εταιρίας, και αυξάνοντας την συνολική παραγωγικότητα και τα επίπεδα εξυπηρέτησης των πελατών της.

Abstract

In this dissertation a basic case of the Vehicle Routing Problem (VRP) is studied, in which a single vehicle starts from its depot and serves customers in a *predefined sequence*. The objective is to serve all customers and minimize travel distance (cost). This problem is of significant practical interest; indicative applications include Ex-van sales and Material Handling systems. Several cases of this problem, of increasing complexity, are posed, analyzed and solved. These cases are:

- Multiple product delivery with deterministic customer demand. Two sub-cases are studied: a) the compartmentalized load and b) the unified load case. The mathematical models, as well as new efficient algorithms that solve these problems to optimality have been developed and analyzed.
- Multiple product delivery with stochastic customer demand. Both sub-cases mentioned above are studied. For both cases we present the characteristics of the respective problems, novel methods to determine the minimum expected cost, and the theoretical results that permit one to determine the optimal decision after serving each customer. Both cases have been addressed using dynamic programming, and for both it has been proven that there exists an appropriate threshold function for each customer, which can be used to determine the optimal decision. Extensive analysis of the proposed algorithms has been conducted.
- Pickup and delivery (of product) with stochastic customer demands. In this case the vehicle not only delivers products to customers but it also picks up returned items from each customer (e.g. damaged goods, or empty packaging). The characteristics of the problem have been presented, together with a novel method to determine the minimum expected cost, and the optimal decision after serving each customer. The proposed method has also been analyzed extensively.

This work may support a decision support framework, which can be utilized in fixed routing operations for a wide variety of cases and applications (deterministic or stochastic demand, single or multiple products, delivery or pickup & delivery): Thus, ad-hoc sub-optimal decisions can be eliminated, minimizing total operating costs, and increasing the overall productivity and customer service of the distribution fleet.

PhD Publications

Journal Publications

- Tsiribas, P., Tatarakis, A., Minis, I., Kyriakidis, G. (2007) “Single vehicle routing with multiple depot returns”, European Journal of Operational Research, Vol. 187, No. 2, pp. 483-495.

- Zeimpekis, V., Tatarakis, A., Giaglis, G. M., Minis, I. (2006) “Towards a Dynamic Real-Time Vehicle Management System for Urban Distribution”, International Journal of Integrated Supply Management, (In Press).

- Giaglis, G. M., Minis, I., Tatarakis, A., Zeimpekis, V. (2004) “Minimizing Logistics Risk through Real-Time Vehicle Routing and Mobile Technologies: Research To-Date and Future Trends”, International Journal of Physical Distribution and Logistics Management, Vol. 34, No. 9, pp.749-764.

Conference Publications

- Zeimpekis, V., Giaglis, G., Tatarakis, A. (2004) *“*A systemic approach to real-time vehicle re-routing for urban distributions”, *In the proceedings of the 20th European Conference on Operational Research, (EURO XX)*, 4-7 July, Rhodes, Greece.

- Giaglis, G. M., Minis, I., Tatarakis, A. Zeimepkis, V. (2003) “Real-time Decision Support Systems in Urban Distributions: Opportunities afforded by mobile and wireless technologies” *In the proceedings of 3rd International ECR Research Symposium*, 11-12 September, Athens Greece.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

One of the areas of Operation Research that has attracted significant research attention at least for the past 30 years is the area of optimizing the transportation and distribution of goods. In an urban environment distribution represents, on average, the highest portion of logistics costs (Ballou, 1999). In some special cases, such as the beverage or the alcoholic drinks industry, distribution costs account for 70% of the value-added activity costs (Golden and Wasil, 1987). It, therefore, becomes apparent that cost effectiveness of transportation and distribution is of outmost importance for the competitiveness of this and other sectors. The main parameters that influence transport and distribution activities, and, therefore, affect costs directly include:

- Transport / Distribution Network (network architecture, number and location of facilities such as hubs and depots, the number and location of destination customers, or sites, etc.)
- The number and capacity of vehicles in the fleet
- Service parameters (delivery unit loads, time windows, etc.)
- Operational parameters (shifts, activity plans, etc)
- Fleet planning – routing.

This dissertation focuses on the last area, and especially on vehicle routing. The design of optimal or near optimal delivery routes, in the case in which distribution vehicles originate from a central depot and serve a number of customer points, is generally referred in the bibliography as the Vehicle Routing Problem (VRP) (Toth and Vigo, 2002b). When solving the VRP, it is common in the literature to assume that the customer demand is known in advance. This assumption is valid in a range of practical cases, in which delivery is performed based on fixed customer orders. Many algorithms have been proposed to address these cases (Dantzig and Ramser, 1959; Clarke and Wright, 1964; Assad, 1988; Golden & Assad, 1988; Laporte and Osman, 1995; Toth and Vigo, 2002b). However, in other cases, the customer demand may be random, and/or other parameters may be stochastic (e.g. travel time / cost of the network arcs). Routing problems that involve randomness are characterized as Stochastic Vehicle Routing Problems (SVRP).

The Vehicle Routing Problem with Stochastic Demands (VRPSD) belongs to a category of *a priori* optimization problems (Bertsimas *et al*, 1990) for which it is impractical to consider an *a posteriori* approach (according to which an optimal solution is recomputed every time the value of a stochastic demand is revealed). Instead, an *a priori* solution attempts to obtain the best available solution over a range of problem scenarios, prior to the realization of any single scenario. According to Roberts and Hadjiconstantinou (1998), who evaluated the computational performance of both types of solution methods, the *a priori* solution of a Vehicle Routing Problem with random demand resides, on average, within 8% of the solution obtained by a reoptimization-based, *a posteriori* strategy.

In this dissertation a basic case of the VRP is studied; in this case, a single vehicle starts from its depot and serves customers in a *predefined sequence*. The demand of each customer is either known in advance (deterministic) or not (stochastic). The distances (travel times or costs), among all points of the network (depot and customer points) are fixed and known. The quantity to be loaded to the vehicle cannot exceed its capacity. Upon completion of service at each customer site, the vehicle has to either (a) travel to the next customer, or (b) return to the depot in order to reload (and/or unload), and resume its route. Note that even though the stock on board the vehicle may be adequate to serve the next client, a preemptive return to replenish the vehicle's stock may be beneficial in order to avoid future returns from a customer site that is further away from the depot. The objective is to serve all customers and minimize travel distance (cost). The predefined customer visit sequence is a significant restriction, which, nevertheless is encountered in many practical cases (due to hard customer time windows, traffic avoidance practices, etc.).

For the single product case both the deterministic and the stochastic version of this problem have been addressed in the literature (see Yang *et al.* (2000)). In this dissertation more complex cases are studied, which also present significant practical value. These cases are:

- Multiple product delivery with deterministic customer demand. Two sub-cases are studied: a) the compartmentalized load and b) the unified load case.
- Multiple product delivery with stochastic customer demand. Both sub-cases mentioned above are studied
- Pickup and delivery (of product) with stochastic customer demands.

All five cases above are of significant practical value in the Logistics industry (e.g. Ex-van Sales) and in material handling within manufacturing plants with fixed route vehicles. The remainder of this dissertation is structured as follows: Chapter 2 presents the most relevant research to-date in this field. Furthermore, in this chapter we identify the research gaps, we define the problems to be addressed, and discuss the theoretical value and practical implications of these problems.

In Chapter 3 we present the basic form of the Vehicle Routing with Depot Returns Problem (VRDRP) under deterministic demand. The problem characteristics, the mathematical model and an optimal solution method is described. This method is based on dynamic programming and has been inspired by the work of Yang *et al.* (2000).

In Chapter 4 we propose and investigate: (i) the case of multiple-product deliveries in which each product is stored in its own compartment in the vehicle and (ii) the case of multiple-product deliveries in which all products are stored together in the vehicle's single compartment. The problem characteristics, the mathematical models and optimal solution methods are described. The Chapter concludes with performance analysis of the solution method.

In Chapter 5 we present the stochastic version of the Vehicle Routing with Depot Returns Problem (SVRDRP). In this problem the customer demands are assumed to be independent random variables with known distributions. The purpose of this chapter is to analyze the problem with respect to critical parameters that characterize the randomness of the demand. This lays the foundation for considerable enhancements in the next chapters.

In Chapter 6 we extend the Stochastic Vehicle Routing with Depot Returns Problem (SVRDRP) to address the case of distributing multiple product types. In line with Chapter 4 we address two cases; compartmentalized and unified load. In this chapter we present the characteristics of each problem, new methods to determine the minimum expected cost, and the theoretical results that permit us to determine the optimal decision after serving each customer. The performance of the proposed methods is analyzed by solving a large number of sample problems per case.

In Chapter 7 we examine the Pickup and Delivery case of the VRDRP under random demand. The characteristics and the mathematical formulation of the problem are presented, together with a new method to determine the minimum expected cost. Furthermore the proposed method is used to determine the optimal decision after serving each customer. Finally, the performance of the proposed methods is analyzed by solving a large number of sample problems.

The dissertation concludes with Chapter 8, in which the contribution of this dissertation and the related conclusions are presented, together with future research directions.

Chapter 2

Background

2.1 Introduction

This Chapter presents the main contributions of the literature that are directly related to the problem(s) studied in this dissertation. We overview the basic types of goods distribution in an urban environment, and define the related problems addressed in this work. For these problems, we review the most relevant research to-date and identify promising areas for further work. The Chapter concludes by discussing new contributions of the present dissertation in these areas.

2.2 The Urban Distribution Environment

The Urban Distribution Environment represents one of the most complex settings of operations for a distribution company. Congested road networks, one-way systems, traffic peaks at particular times and areas are some of the complexity factors that characterize this setting.

One may distinguish at least two ways of distributing goods in urban areas: *Standard deliveries* and *Ex-van sales* (Giaglis *et al.*, 2004). While in both cases operations are performed within a typical delivery network with *N* warehouses and *M* customers, that are served by a fleet of *K* vehicles, these two cases differ in the way they handle demand. In standard deliveries the demand is known (usually driven by pre-placed customer orders), while Ex-van sales operate in an unknown demand environment, in which orders are placed during delivery at the customer site. Table 2.1 summarizes the main attributes of the two modes of urban deliveries.

Table 2.1. Characteristics of Standard Deliveries vs. Ex-van Sales in Urban Distribution

Standard Deliveries	*Ex-van Sales*
Fixed geographical Network	
Fixed truck and fleet capacity	
Known demand per customer site	Unknown demand per sales point
Fleet delivers based on orders	Orders are not known in advance (only an estimate of aggregate demand in the sales area exists)
Fixed schedules and delivery time windows	More relaxed schedules and delivery time windows
Truck routes determined *a priori* based on demand, network traffic, and other parameters in a near-optimal way	Distribution of work per truck is based on past area sales and business agreements with the drivers

A significant amount of research has focused on standard deliveries in the past. Ex-van Sales have not received as much attention, and it is this type of Urban Distribution that forms the motivation for the work of this dissertation.

2.3 The Vehicle Routing Problem

Transportation and distribution contribute approximately 20% to the total costs of a product (Reimann *et al*., 2003). These costs incur between any two subsequent links of the supply chain, and between the final link and the end customer. Both industry and academia have long recognized the potential for optimization of operations in this area. Formally, most problems in goods distribution are related to the Vehicle Routing Problem (VRP). This problem is a generalization of the classic Traveling Salesman problem (TSP) (Christofides, 1979; Cornuejols and Nemhauser, 1978; Gendreau *et al*, 1997), and seeks a set of efficient vehicle routes to serve a number of geographically dispersed customers. The VRP was introduced by Dantzig and Ramser (1959), almost 50 years ago; in this original work they described a practical application concerning the delivery of gasoline to service stations, and proposed the first mathematical programming formulation and algorithmic approach to solve it. Since then, the VRP has received considerable research attention, and has become one of the fundamental problems of Operations Research.

The objective of the VRP is to deliver goods to a set of customers with known demands following minimum-cost vehicle routes originating from and terminating at a depot (Clarke and Wright, 1964; Assad, 1988; Golden & Assad, 1988; Laporte and Osman, 1995). A very useful survey of significant research results in this problem is given by Toth & Vigo (2002b).

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

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

$$\text{subject to} \sum_{i,j} \mu_i x_{ijk} \le Q \qquad k = 1,2,\ldots,m$$

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

where:

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

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

m = the number of vehicles available

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

μ_i = the amount demanded at location i

Q = the vehicle capacity.

From the above formulation it is clear that the VRP is an integer-programming problem. It is also an NP-hard problem (for information about the theory of NP-completeness refer to Garey and Johnson 1979), and, therefore, practical problem instances cannot be solved to optimality within reasonable time; in fact there are no exact algorithms available that consistently solve problems with more than 50–75 customers (see Toth & Vigo, 2002).

2.3.1 Modeling approaches for the VRP

According to Toth & Vigo (2002b), three basic modeling approaches have been proposed in the literature for the VRP. The models of the first type are known as *vehicle flow formulations* and they use integer variables associated with each arc or edge of the graph (modeling the distribution network), which count the number of times the arc or edge is traversed by a vehicle. These are the most frequently used models for the basic versions of the VRP; they are particularly suited for cases in which a) the cost of the solution can be expressed as the sum of the costs associated with the arcs, and b) the most relevant constraints concern the direct transition between the customers within the route, so they can be effectively modeled through an appropriate definition of the arc set and the arc costs. On the other hand, vehicle flow models cannot be used to handle some practical issues, such as in cases in which the cost of a solution depends on the overall vertex sequence, or on the type of vehicle assigned to a particular route (Toth & Vigo 2002b). The second family of models is based on the so-called *commodity flow formulation*. In this type of model, additional integer variables are associated with the arcs or edges and represent the flow of commodities along the paths traveled by the vehicles. Only recently have models of this type been used as the basis for the exact solution of Capacitated VRP (CVRP).

The models of the third family have an exponential number of binary variables, each associated with a different feasible circuit. The VRP is then formulated as a *Set-Partitioning Problem* (SPP) seeking a collection of circuits that minimize cost, serving each customer once and possibly satisfying additional constraints. A main advantage of this model is that it allows for extremely general route costs (for modeling costs that depend on the sequence of arcs and/or on the vehicle type). Moreover, the additional side constraints do not need to take into account restrictions concerning the feasibility of a single route. As a result, the constraints can often be replaced with a compact set of inequalities. This produces a formulation, the linear relaxation of which is typically much tighter than that of the previous model types (Toth and Vigo, 2002b).

2.3.2 Extending the fundamental VRP

In an effort to take into consideration important practical issues, the fundamental VRP has been extended in a number of aspects. Indeed, one can distinguish several issues of practical importance that raise considerable challenges in VRP-related research (Giaglis *et al.*, 2004).

Vehicle Capacity**:** There exist formulations for both the *Capacitated VRP* and the *Uncapacitated VRP* depending on whether vehicle capacities are considered. The Capacitated VRP (CVRP), as presented for example in Toth and Vigo (2002a), is perhaps amongst the most widely researched variations of the problem. Capacity considerations are important in the case examined here, especially in view of reverse logistics, in which the capability of the vehicle to respond to the customer requirements depends strongly on its available capacity.

Number of Stages**:** While the *single-stage VRP (delivery only)* is primarily concerned with the establishment of outbound delivery routes, the *double-stage VRP* considers both *delivery & pickup*, i.e. outbound and inbound distribution. For a treatment of the two-stage VRP see Savelsbergh (1995) and Yang *et al*. (2000).

Deterministic vs Stochastic Supply/Demand**:** The *Deterministic VRP* assumes that demand/supply is known *a priori*, while the *Stochastic VRP* encompasses uncertainty in demand and/or supply levels (Min *et al.,* 1998). As discussed above, demand uncertainty is a key characteristic of Ex-van sales (see Section 2).

Planning Horizon (single/multiple periods)**:** The *Single Period VRP* takes into consideration a single planning period (for example, solving the distribution problem for next day's deliveries), while the *Multiple Period VRP* considers optimal solutions in multiple periods and therefore seeks for a good solution over a longer planning horizon. In this case the initial schedule can be adjusted, according to the current needs for distribution (Laporte, 1988).

Time Windows**:** A classical variation of the VRP considers time windows, outside which deliveries cannot be accepted. Time windows can either be 'hard', when they cannot be violated, or 'soft', in which case violations are accepted but penalized. A recent analysis of the VRP with soft time windows has been provided by Ioannou *et al.*(2003).

Objectives**:** There exist *Single-Objective* or *Multiple-Objective* formulations of the VRP. The most common VRP objective is to minimize the total cost of deliveries. However, additional objectives might be considered, such as minimizing number of depots or maximizing customer satisfaction (Renauld *et al.*, 2000; Fisher, 1994).

Type of Approach: The computational complexity of the VRP has prompted the development of heuristics since the 1970s (Christofides *et al.*, 1969; Yellow, 1970; Wren and Holliday, 1972; Ashour *et al.*, 1972; Gillett and Miller, 1974). The development of heuristics especially in practical VRP cases still comprises a significant research area (Laporte, 1992; Breedam, 1995; Hachicha *et al.,* 2000; Laporte *et al.,* 2000). *Exact* solutions have also been developed; however, they can only be applied to vehicle routing problems of limited complexity (Reimann, 2003). An example of an exact, branch-and-bound approach is presented by Fisher (1994), in which the solution approach uses the minimum k-tree approach.

Table 2.2 (Giaglis *et al*, 2004) includes relevant VRP publications (the majority of which have been mentioned above) and indicates that, while specific cases of the VRP have been rather extensively addressed in the literature, others have not attracted similar attention. For example, a relatively limited number of publications have focused in topics, such as the double-stage delivery, stochastic demand/supply, time-windows, and multiple objectives. At the same time, more than approximately two-thirds of the approaches employed use heuristics, while exact approaches can be found in about one-third of the cases.

It is also worth pointing out that the problem of Ex-van sales, which incorporates several complexities, such as uncertain demand, multiple planning horizons, possible time windows, and others, has yet to be fully addressed in the literature, despite being an important practical case with significant potential for improvement.

Table 2.2. VRP Taxonomy

	Number of Stages		Nature of demand/supply		Vehicle capacity		Planning horizon		Time windows			Objectives		Type of Approach	
	Single stage	Double stage	Deterministic	Stohastic	Uncapacitated	Capacitated	Single period	Multiple periods	No time Windows	Soft windows	Hard windows	Single objective	Multiple objectives	Exact	Heuristic
Ashour *et al* (1972)	■		■		■		■		■			■			■
Gillett and Miller (1974)	■		■			■	■		■			■			■
Dror and Ball (1987)	■			■	■		■		■			■			■
Robuste *et al* (1990)	■			■	■		■		■			■			■
Vliet *et al* (1992)	■		■			■	■				■		■		■
Fisher (1994)	■		■			■	■		■			■		■	
Koksalan *et al* (1995)	■		■			■	■		■			■		■	
Savelsbergh (1995)		■	■			■		■	■			■		■	
Bowers *et al* (1996)	■		■			■	■		■			■			■
Beasley and Crhistofides (1997)	■		■		■		■		■			■		■	
Modesti and Sciomachen (1998)	■			■	■		■		■				■	■	
Barbarosoglu and Ozgur (1999)	■		■		■		■		■			■			■
Van der Poort *et al* (1999)	■		■		■		■		■			■		■	
Coy *et al* (1999)	■		■		■		■		■			■			■
Larsen (1999)	■		■		■		■				■	■			■
Secomandi (2000)	■			■		■		■	■			■		■	
Renaud *et al* (2000)	■		■			■	■		■			■			■
Nanry and Barnes (2000)	■		■			■	■				■	■			■
Yang *et al* (2000)		■		■		■		■		■		■			■
Rego (2001)	■		■			■	■		■			■			■
Fagerholt (2001)	■		■			■	■			■		■			■
Glover *et al* (2001)	■		■		■		■		■			■			■
Toth and Vigo (2002a)	■		■		■		■		■			■		■	
Tarantilis and Kiranoudis (2002)	■		■		■		■		■			■			■
Tarantilis *et al* (2003)	■		■			■	■		■			■			■
Ho and Haugland (2003)	■		■			■	■				■	■			■
Ioannou *et al* (2003)	■		■			■	■			■		■			■
Coverage	92%	8%	77%	23%	42%	58%	88%	12%	73%	12%	15%	92%	8%	31%	69%

2.4 The Ex-van Business Model

Typically retail outlets (supermarkets, kiosks) monitor stock levels per item, and record seasonal consumption trends. Based on this information they compile a forecast, which, combined with actual stocks, is translated to purchase orders containing the quantity to be purchased per product. The lead-time of purchase order processing (internal), as well as the lead-time of the supplier to respond to the order and dispatch the goods (external), forms the total lead-time of such an order.

There are typically two sources of variability that may limit the effectiveness of outlet supply: First, there is lead time variability which may be caused by unforeseen events and warehouse stock-outs. Secondly, there is demand variability, and inevitable deviations from the forecast. These types of variability may affect the ability of the outlet to satisfy the total daily customer demand for certain items. The Ex-van model attempts to respond to such variations in an effective manner targeting especially high-demand commodities. This is done by replenishing regularly the stock of certain types of commodities, so that the outlet (super market, kiosk, etc.) can maximise its sales.

The typical Ex-van commodity types vary and depend on the type of outlet. For example, Ex-van commodities in a Super-Market present the following characteristics:

- High daily customer demand
- Short expiration dates
- High storage requirements
- Low value per unit item.

Typical examples include fresh milk cartons, fresh yogurt, fresh fruit juice and fresh bread. Therefore, the retail outlet may order a specific quantity per item via standard delivery, but also have a scheduled Ex-van visit around mid-day. At the time of the visit the particular commodity stock levels will be examined, and if found below a predefined threshold per commodity (assigned in advance by the outlet Inventory Manager), additional product will be purchased from the Ex-van vehicle, therefore creating an Ex-van sales order. The size of the Ex-van sales order may vary, depending on daily sales as well as the sales skills of the Ex-van driver / sales person. The typical Ex-van fleet size for commodities similar to the ones discussed above is between 5-10 vehicles. Each vehicle is assigned a set of customers to be visited on a daily basis. Due to the fact that each customer can define a time window within which Ex-van visits are allowed, the customer service sequence is typically predefined and has to be observed, otherwise sales opportunities may be lost.

The mission of each Ex-van vehicle is to visit all assigned customer sites, and replenish the stock of selected products. The demand of each customer point is not known in advance but it is revealed upon arrival. Therefore, the total demand of a scheduled customer sequence typically exceeds the total capacity of the vehicle for a particular item, forcing the vehicle to return to the depot in order to replenish its own stock, before resuming its route.

As mentioned above, the Ex-van driver is a mobile Sales person, who satisfies the customer (pull model), and attempts to sell as much as possible to the customer (push model). It is frequently the case, that the outlet's actual demand may be less than the actual sales quantity, due to the sales-orientation of the Ex-van driver, an *up*-sell case. It is also common for the outlet to demand commodity *A* but to finally purchase both *A* and *B*, an example of a *cross*-sell case. The Ex-van driver strives to maximise sales per customer point, since there is typically a direct relation between the sales achieved and the commission / bonus of the 'driver'.

In summary, the characteristics of the Ex-van Sales Model are:

a) The distribution vehicle operates in a designated area
b) The sequence of serving the customers within this area is typically predetermined, in line with the time-window constraints of these customers
c) The vehicle usually carries multiple items
d) The customer demand is not known in advance
e) The vehicle may pickup returned packaging or expired products and carry them to the depot
f) If the vehicle disperses its entire inventory (of one or more items) prior to completing the route, it returns to the depot for stock replenishment.

Other Related Settings

In addition to the Ex-van environment, the above characteristics may also arise in other practical settings. For example, material handling systems in a manufacturing shop often operate along fixed pathways that connect the material warehouse with workcenters located along this pathway. Consider the case of Automated Guided Vehicle Systems (AGVs), which are self-propelled vehicles typically guided along a magnetic induction strip, or a painted strip on the shop floor, and transport discrete parts to workcenters, obviously in a predefined sequence. Note that in addition to the main pathway connecting the workcenters, there are spurs connecting each workcenter with the material warehouse, allowing the return and reloading of the AGV. This case shares the same characteristics with the Ex-van sales case: The AGV serves a specific area on the shop-floor, has limited capacity, can carry multiple items, the visit sequence is predetermined, items may also be picked up from the workcenters, the demand of each workcenter may not be known in advance (especially for items to be picked up), and the AGV is allowed to return to the material warehouse for stock replenishment. It should also be mentioned that a similar situation exists in other types of material handling mechanisms, such as monorail or powered overhead conveyors, which carry parts on a hanger with limited capacity.

2.5 Relevant VRP problems

The types of vehicle routing problems that are most relevant to the setting described above and to the work of this dissertation are presented in Table 2.3 and discussed below.

Table 2.3. Relevant VRP types.

Problem Type	Characteristic
Capacitated VRP	Each vehicle has a limited capacity
Split Delivery VRP	The customers are served by *n* vehicles
VRP with Pickups and Deliveries	The vehicle delivers products to customers and collects items from them
Stochastic VRP	Some of the problem parameters or variables are not known in advance

2.5.1 The Capacitated and Split Delivery Problems

In the *Capacitated VRP* (CVRP), the demands are deterministic, known in advance, the delivery vehicles are identical (of equal capacity) and are based at a single central depot, and the objective is to minimize the total cost (i.e., the total distance or travel time) needed to serve all customers. Generally, the travel cost between each pair of customers is the same in both directions, i.e., the resulting cost matrix is symmetric. In some applications, such as distribution in urban areas with one-way streets, the cost matrix may not be symmetric. The CVRP has been extensively studied since the early sixties and, as a result, many heuristic and exact approaches have been proposed (Laporte and Louveaux, 1990; Augerat *et al.* 1998; Toth and Vigo, 2002; Tarantilis *et al*, 2005; Longo *et al.* 2006, Alba and Dorronsoro, 2006).

The *Split Delivery* VRP (SDVRP) is a relaxation of the VRP according to which the same customer is allowed to be served by different vehicles, if this reduces overall costs. This relaxation is very important if the size of the customer orders is in the order of the capacity of a vehicle. Dror & Trudeau (1990) have proposed a heuristic algorithm for the SDVRP and have shown that allowing split deliveries can yield substantial savings, both in the total distance traveled and in the number of vehicles used in the optimal solution. In addition, the SDVRP has been formulated as an integer linear program and its solution has been approached by constraint relaxation branch and bound algorithms (Dror *et al.* 1994; Ho & Haugland, 2004; Bompadre *et al*, 2006).

2.5.2 The VRP with Pickups and Deliveries

In the Vehicle Routing Problem with *Pickups and Deliveries* (VRPPD), the customers may also return some items during the vehicle's visit (e.g. empty packaging, returned product to be delivered to the depot, items to be delivered to another customer). In the basic version of the VRPPD, each customer i is associated with two quantities d_i and p_i representing the demand of commodities (measured by the same unit of measure) to be delivered and picked up at customer i, respectively. For each customer i, O_i denotes the vertex that is the origin vertex of the delivery demand, and D_i denotes the destination vertex of the pickup demand. It is assumed that at each customer location the delivery is performed before the pickup; therefore the current load of a vehicle arriving at a given location is defined by the initial load minus all products already delivered plus all products already picked up. The VRPPD consists of finding a collection of exactly K simple circuits with minimum cost, such that:

- Each circuit visits the depot vertex;
- The current load of the vehicle along the circuit must be nonnegative and may never exceed the vehicle capacity Q;
- For each customer i, the origin O_i, when different for the depot, must be served in the same circuit and before customer i; and
- For each customer i, the destination D_i, when different from the depot, must be served in the same circuit and after customer i.

It is, therefore, obvious that in the VRPPD it is necessary to plan for maintaining enough empty space on the vehicle in order to accommodate the returned goods or items (in cases in which deliveries and pick ups are of the same order). This restriction makes the planning problem harder and can lead to sub-optimal utilization of vehicle capacities, increased travel distances or a need for additional vehicles. The VRPPD is NP-hard in the strong sense, since it generalizes the Capacitated VRP (CVRP). The latter is obtained when $O_i = D_i$ and $p_i = 0$ for each $i \in V$.

Variants of the VRPPD include the so-called TSP with Pickup and Delivery (TSPPD), in which $K = 1$. In a significant common variant, all delivery demands start from the depot and all pickup demands are brought back to the depot, and, thus, there are no interchanges of goods between customers. Other problem variants include; a) relaxing the restriction that all customers have to be visited exactly once, or b) each vehicle must deliver all the commodities before picking up any items.

The solution of the VRPPD has recently been approached by promising metaheuristics which include tabu-search using arc-exchange-based and node-exchange-based neighborhoods, and employing different and interacting tabu lists (Righini, 2000; Nagy & Salhi, 2005; Alfredo *et al.*, 2006; Pisinger and Ropke, 2007).

2.5.3 The Stochastic VRP

The Stochastic Vehicle Routing Problem (SVRP) refers to a family of problems, that combine the characteristics of stochastic and integer programs, and are often regarded as computationally intractable (Gendreau *et al*, 1996). Therefore, only relatively small instances can be solved to optimality and effective heuristics are hard to design and assess. The most common stochastic VRPs are the Vehicle Routing Problem with Stochastic Demands (Dror *et al.*, 1989; Laporte, 1989), the VRP with Stochastic Customers (Bertsimas, 1988; Waters, 1989), and the VRP with Stochastic Customers and Demands (Jezequel, 1985; Jaillet, 1987; Jaillet and Odoni, 1988).

The VRP with Stochastic Demands (VRPSD)

In this problem, the demands are usually independent random variables that may (or may not) follow a known distribution based on historical demand. This problem often arises in practice. A typical example is garbage collection, in which it is impossible to know *a priori* the quantity to be collected at each collection. Another example, is the delivery of petrol to petrol stations. In this case, when a customer issues an order it is unknown how much petrol will be sold in the time elapsed between the order and the delivery.

In order to address the inherent uncertainty in this type of VRP, a recourse action (i.e. return to the depot in order to refill) is usually embedded into the formulation of the problem, and penalties are incurred in the case of a route failure (Stewart and Golden, 1983). Due to the stochastic nature of this problem, the objective function is the expected value of the total route cost; the goal is to approach the optimal value, which can be derived from the deterministic counterpart of the particular problem (Bertsimas, 1992; Trudeau & Dror, 1992; Dror, 1993; Birattari *et al.* 2005). Bertsimas (1992) constructs an *a priori* sequence among all customers of minimal expected total length and proposes heuristics for the solution of the problem. His approach proved to be a strong and useful alternative to the strategy of re-optimization in capacitated routing problems. More recently, Birattari *et al.* (2005), proposed five metaheuristics (simulated annealing, tabu search, local search, ant colony optimisation, and evolutionary algorithms) and tested the effect of hybridization (TSP-*approximation* and VRPSD-*approximation*).

Secomandi (2000, 2001) has fairly recently applied Neuro Dynamic Programming techniques to the VRPSD. He addressed the VRPSD with a re-optimization approach, in which after each customer demand is revealed the remaining part of the problem is re-solved. This approach may yield better solutions than the preventive restocking strategy (returning to the depot before a stock out actually occurs), but it is much more computationally expensive. Moreover, the initially planned route may be altered completely, and this situation may present a limitation in practice.

Yang *et al.* (2000) investigate the single and multi-vehicle VRPSD. Instead of adopting a simple recourse action usually suggested in the literature, an optimal restocking policy of the vehicle has been incorporated in the route design. In particular, the restocking points are deliberately planned along the route, such that the probability of the route failure, and the accompanying recourse cost (including any penalty) is reduced, and the expected total cost of the routes is minimized. Two heuristic algorithms are developed to construct both single and multiple routes that minimize total travel cost. The algorithms (route-first-cluster-next, and cluster-first-route-next) solve separately a) the problem of clustering customers to be served by different vehicles, and b) the problem of finding the best route within each cluster. Both algorithms seem to be efficient and robust for small size instances, as shown by comparing the results to those obtained from branch-and-bound solutions for instances with up to 15 customers. It has been shown that, for the unconstrained case, a single route design gives the best solution. However, for many practical situations, a large route is impractical, due to various practical restrictions.

The VRP with Stochastic Customers (VRPSC)

In this problem the customers are present in the route with some probability but they have deterministic demands. The vehicle’s total capacity must be respected and returns to the depot may become necessary, if the total route demand exceeds the vehicle capacity. According to Gendreau *et al.* (1996), two interesting properties stand out and apply both to the VRPSD and the VRPSC. First, even if travel costs are symmetrical, the overall solution cost depends on the direction of travel (Dror and Trudeau, 1986; Jaillet and Odoni, 1988). Dror and Trudeau (1986) present two stochastic programming models: Chance-constrained programming models and dependent-chance programming models. A genetic algorithm is designed for solving the proposed stochastic programming models, and the effectiveness of this algorithm is illustrated by solving numerical examples.

The VRP with both Stochastic Customers and Demands (VRPSCD)

In this problem the customers are present in the route with some probability and their demands are also independent random variables. The VRPSCD is an exceedingly difficult problem. Even computing the value of the objective function is complex. Bertsimas (1992) provides a recursive expression for the VRPSCD, as well as bounds, asymptotic results and an analysis of several re-optimization policies. Motivated by applications in strategic planning and distribution systems, rather than resolving the problem when the demand becomes known, Bertsimas proposes to construct an *a priori* sequence among all customers of minimal expected total length, and proposes heuristics for the solution of the problem (under general probabilistic assumptions). His approach proves to be a strong and useful alternative to the strategy of re-optimization in capacitated routing problems.

2.6 The Vehicle Routing with Depot Returns Problem

The problem that models the case(s) discussed in Section 2.4 will be termed the Vehicle Routing with Depot Returns Problem (VRDRP). In this problem, a single vehicle starts from its depot and serves customers in a *predefined sequence*. The demand of each customer is either known in advance (deterministic) or not (stochastic). The distances (travel times or costs), among all points of the network (depot and customer points) are fixed and known. The quantity to be loaded to the vehicle cannot exceed its capacity. Upon completion of service at each customer site, the vehicle has to either (a) travel to the next customer, or (b) return to the depot in order to reload (and/or unload), and resume its route. Note that even though the stock on board the vehicle may be adequate to serve the next client, a preemptive return to replenish the vehicle's stock may be beneficial in order to avoid future returns from a customer site that is further away from the depot. The objective is to serve all customers and minimize travel distance (cost).

The predefined customer visit sequence is a significant restriction, which, nevertheless is encountered in many practical cases (due to hard customer time windows, traffic avoidance practices, etc.). One may define several cases of this problem of increasing complexity. These cases are discussed below, along with the available literature.

The deterministic VRDRP

According to this problem, the demand of each customer is known in advance (deterministic) and the challenge is to identify the minimum cost route, which may include depot returns for vehicle stock replenishment. This problem is addressed in Chapter 3 of this dissertation The VRDRP has distinct differences when compared to the classic Vehicle Routing Problem (VRP):

1. The customer visit sequence in the VRDRP is pre-determined.
2. In the VRDRP the vehicle may visit the depot multiple times. In fact, it may be advantageous to return to the depot, even if the stock on board is adequate to satisfy the demand of the next customer(s), if such a preemptive return minimizes the total route distance.
3. The VRP yields a solution in which the number of routes (tours) is equal (or less in some extreme cases) to the number of vehicles. The VRDRP yields solutions with a number of tours less or equal to the number of customers.

VRDRP with Multiple Products

In this case multiple products are delivered to the customers. The demand of each customer for each product is known in advance (deterministic). The challenge is again to identify the minimum cost route including the necessary depot returns for vehicle stock replenishment. This case consists of two distinct sub-cases, a) the sub-case in which each product is stored in its own compartment the capacity of which is fixed, and b) the sub-case in which all products are stored together in the vehicle's single compartment.

Note that, in this last (unified) load sub-case there are additional issues to be considered; for example, an additional decision needs to be made regarding the quantities of stock to be loaded onto the vehicle. This problem is addressed in Chapter 4 of this dissertation.

VRDRP with Pickups and Deliveries

In this case the demand of each customer is known in advance (deterministic) but it involves both delivery as well as pickup of goods. Thus, an additional decision should be made concerning the quantity to be loaded to the vehicle each time the vehicle returns to the depot. This is because unnecessarily high stock levels may prevent the collection of returned items, therefore causing additional depot returns and lower customer service. The challenge is again to identify the minimum cost route. This problem has been presented and addressed in Tsiribas *et al.* (2007).

The Stochastic VRDRP

In this case, a single product is distributed, but the demand of each customer is not known in advance. The objective is again to identify the minimum expected cost including the necessary depot returns for vehicle stock replenishment. This problem was initially treated in Yang *et al.* (2000). In this paper, the authors propose two heuristic algorithms to construct both single and multiple routes that minimize total travel cost. These methods were found to be, in general, superior to those of which adapt a deterministic method. The two significant results of Yang *et al* that relate to our work are: a) The dynamic programming formulation to determine the minimum expected cost, which has been extended in this work to address more complex and intersting scenarios, and b) the theory of deriving optimal policy of service.

The stochastic VRDRP was more recently presented by Manfrin *et al.* (2004), who approached it as a simplified version (1 instead of *n* vehicles) of the generalized Stochastic Vehicle Routing Problem (SVRP). In their paper, Manfrin *et al.* explore the hybridization of the metaheuristic search process by interleaving the objective function with the one from a closely related problem (the traveling salesman problem - TSP) which can be computed in much less computation time. Moreover, Manfrin *et al.* analyze several extensions to the proposed metaheuristics, and report experimental results with respect to different types of instances. It is shown that for the instances tested, most metaheuristics perform better when hybridized with the traveling salesman objective function. Lately, Kyriakidis and Dimitrakos (2007), presented the SVRDRP with continuous demands, and suggested a dynamic programming algorithm to determine the optimal policy.A detailed treatment of the Stochastic Vehicle Routing with Depot Returns Problem (SVRDRP) is included in Chapter 5 of this dissertation.

The SVRDRP with Multiple Products

This is the stochastic version of the VRDRP with Multiple Products. The challenge is again to identify the minimum expected cost including the necessary depot returns for vehicle stock replenishment. This case includes, again, two distinct sub-cases, as described for the deterministic case. Both sub-cases are modeled and addressed in Chapter 6.

The Stochastic Ex-van with Pickup and Delivery

In this case, neither the demand for the product to be delivered nor for the item to be picked up is known in advance. This is the stochastic version of the VRDRP with Pickup and Delivery. Again, as in the deterministic case, an additional decision should be made concerning the quantity to be loaded to the vehicle each time the vehicle returns to the depot. This is because unnecessarily high stock levels may prevent the collection of returned items, therefore causing additional depot returns and lower customer service. This problem is modeled and solved in Chapter 7.

2.7 Contributions of this Dissertation

As described in the previous Section, this dissertation treats a vehicle routing problem of significant practical value. Several cases of this problem of increasing complexity are modeled and solved. Most of these cases are posed, analyzed, and solved to optimality for the first time in the literature. Our contributions can be summarized as follows:

1. For the VRDRP we develop a dynamic programming algorithm (DPA) inspired by the work of Yang *et al.* (2000) and solve the problem to optimality in efficient computational times. Problem instances containing up to 50 customers were solved.

2. For the two cases of the VRDRP with Multiple Products we develop the mathematical models, as well as new efficient algorithms that solve these problems to optimality. Again, problem instances containing up to 50 customers were solved.

3. For the SVRDRP, which has been initially presented and solved by Yang *et al.* (2000), we present an analysis of the problem to determine: a) the effect of the variance of the demand on the minimum expected cost function, and b) the interaction between the mean and the variance of the demand.

4. For the SVRDRP with Multiple Products we pose both cases, the compartmentalized and the unified load. We analyze the characteristics of each, and we develop novel methods to determine the minimum expected cost. We also develop the theoretical results that permit one to determine the optimal decision after serving each customer.

5. Finally, for the SVRDRP with Pickups and Deliveries we define the problem, present its characteristics, and develop a novel method to determine the minimum expected cost as well as the optimal decision after serving each customer.

The problems in 2, 4 and 5 above, are presented and solved in this dissertation for the first time in the literature.

Chapter 3

The Vehicle Routing with Depot Returns Problem (VRDRP)

3.1 Introduction

In this Chapter we present the basic form of the Vehicle Routing with Depot Returns Problem (VRDRP). In this problem the demand of each customer is known in advance (deterministic). The problem characteristics, the mathematical model and an efficient solution method initially proposed by Yang *et al.* (2000) are presented. The Chapter concludes with comments on the performance of the solution method.

3.2 The VRDRP

As already presented in Chapter 2, the VRDRP is a special case of the VRP in which there is only one vehicle that serves its customers in a predefined sequence delivering a single product. The demand of each customer is known in advance (deterministic). All distances (travel times) in the network are also known. The product quantity that can be loaded to the vehicle may not exceed the vehicle's capacity. The vehicle serves the customer fully during a single visit; i.e. it visits the customer only if the quantity on board is greater or equal to the customer's demand. The vehicle is allowed to return to the depot in order to refill. It is assumed that service at a customer site as well as reloading at the depot, happen instantly.

Upon completion of service at each customer site, the vehicle has to either (a) proceed to the next customer, as long as the demand of the next customer is not greater than the remaining stock on board, or (b) return to the depot in order to reload, and resume its route by visiting the next customer in the sequence. This decision point is shown in Figure 3.1, in which the vehicle has just served *customer-3* and a decision has to be made: in case (a) the vehicle will proceed directly to *customer-4*; in case (b) it will first return to the depot to refill, and will then proceed to *customer-4*. Note that even though the stock on board the vehicle may be adequate to serve the next client, a preemptive return to replenish the vehicle's stock may be beneficial in order to avoid future returns from a customer site that is further away from the depot.

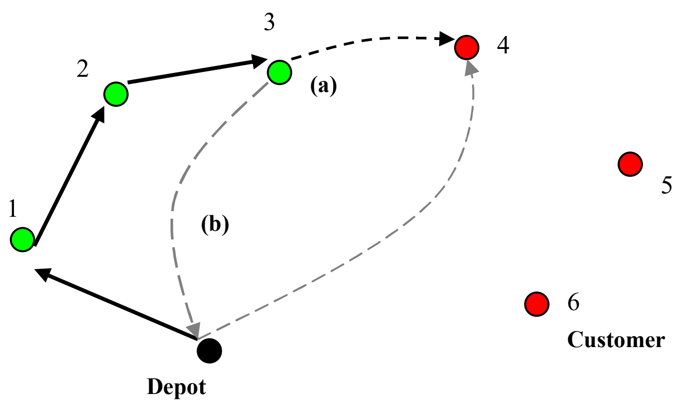


Figure 3.1. The decision the vehicle has to take at each customer point.

Consider a set of nodes $V = \{0, \ldots, n\}$, with node 0 denoting the depot and nodes $1, \ldots, n$ corresponding to customers, and a set of arcs $A = \{(i, i+1), (i+1, 0), (0, i+1) : i \in V - \{n\}\}$ that join the customers along the route $1 \rightarrow 2 \rightarrow \cdots \rightarrow n$, as well as all customers with the depot. The travel cost (distance) of each arc (i, j) is denoted by $c_{ij} > 0$, and the c_{ij} values satisfy the triangular inequality. We assume that a single vehicle must serve all customers according to the predefined sequence $1, \ldots, n$ and that a customer should not be served twice. The vehicle is at the depot initially and after serving all customers it returns to the depot. It is assumed that the maximum capacity of the vehicle is equal to *Q* products. The stock on board the vehicle after serving customer *i-1* is the same with the stock upon arrival at customer *i* (and before serving customer *i*) and equal to z_{i-1}.

Decision Variables:

$x_{i,i+1} \in \{0,1\} \quad \forall \quad i = 1, 2, \ldots, n-1$

$x_{i,0} \in \{0,1\} \quad \forall \quad i = 1, 2, \ldots, n-1$

$x_{0,i} \in \{0,1\} \quad \forall \quad i = 2, \ldots, n$

Objective Function

$$\text{Min } E = \sum_{i=0}^{n-1} c_{i,i+1} x_{i,i+1} + \sum_{i=1}^{n-1} c_{i0} x_{i0} + \sum_{i=2}^{n} c_{0i} x_{0i} \tag{3.1}$$

Constraints

$$x_{01} = 1 \tag{3.2}$$

$$x_{n0} = 1 \tag{3.3}$$

$$x_{i-1,i} + x_{0i} = 1 \quad \forall \quad i = 2, \ldots, n \tag{3.4}$$

$$x_{i,i+1} + x_{i0} = 1 \quad \forall \quad i = 1, 2, \ldots, n-1 \tag{3.5}$$

$$z_{i-1} = x_{0i} Q + (1 - x_{0i})(z_{i-2} - d_{i-1}) \quad \forall \quad i = 2, \ldots, n \tag{3.6}$$

$$z_1 = Q \tag{3.7}$$

$$d_i \leq z_{i-1} \leq Q \quad \forall \quad i = 1, \ldots, n \tag{3.8}$$

Constraints (3.2) and (3.3) indicate that the vehicle must leave the depot at the start of the route and must return to the depot after the completion of the route.

Constraints (3.4) and (3.5) are also network constraints and relate to the customer nodes. According to Eq. (3.4) the vehicle will either arrive to the next customer from the previous customer or from the depot, and according to Eq. (3.5) the vehicle will depart from a customer point either to go to the next customer or to go to the depot.

Constraints (3.6) to (3.8) are capacity constraints. According to Eq. (3.6) the stock left on board prior to visiting customer i will be either equal to the full vehicle capacity Q, if the vehicle is coming from the depot, or equal to what is left on board after serving customer $i-1$. According to Eq. (3.7) the stock of the vehicle upon arrival at the first customer is equal to its full capacity Q. Finally, according to Eq. (3.8) the stock left on board after service completion of a customer, should always be kept between the values of the next customer's demand and the vehicle full capacity. If the value of the stock on board drops below the value of the next customer's demand, a return to the depot for refill is necessary.

3.3 Problem Characteristics

Any feasible route of the vehicle can be denoted by a vector of n elements; each element assumes the values *'0'* or *'1'* : $u_i = 0$ represents the case in which after serving customer i, the vehicle will serve customer $i+1$ without visiting the depot; $u_i = 1$ represents the case in which after serving customer i, the vehicle visits the depot, replenishes its stock, and proceeds to customer $i+1$.

As an example, the vector [0 0 0 1 0 0 0 1] refers to a customer network that consists of eight customer points. After serving customer *4* the vehicle returns to the depot for stock replenishment. Obviously, after serving customer *8* the vehicle returns to the depot in order to conclude its route.

3.3.1 Problem Complexity

If the number of customers in the network is equal to n, then there are 2^{n-1} possible *'0-1'* combinations representing a route, since $u_n = 1$. In order to identify the subset of feasible combinations, the demand-capacity restrictions are applied; that is, the demand of all customers included between two subsequent visits to the depot cannot exceed the total vehicle capacity.

To further analyze the complexity of the problem, we developed an *Exhaustive Search* algorithm that searched the entire solution space exhaustively, identified the feasible combinations, calculated the value of the objective function for each feasible combination, and identified the optimal solution(s). The steps of the Exhaustive Search algorithm are given below:

1. Load the customer matrix C
2. Initialize the total distance traveled as Min = ∞
3. **Do while** all possible combinations have not been exhausted:
    a. Create the first/next combination as [1 0 0 …0]
    b. Examine feasibility of this combination and if feasible
        i. Calculate total distance traveled (Dist)
        ii. If Dist < Min then Min = Dist
        iii. If Dist > Min then Min = Min
    c. If the combination is not feasible go to step (a)
4. End

A *'feasible combination'* is one that the sum of the customer demands among two consecutive depot visits does not exceed the capacity of the vehicle. C is an $n*m$ matrix, where n = number of customers, m = 4. Column one contains information regarding the number of customers in the sequence. Column two contains information on the distance of customer i from the depot. Column three contains information on the distance of customer i from customer $i+1$. Column four contains information on the demand of customer i.

The exhaustive search algorithm was written in *Matlab Version 7.0* and ran on a PC Intel Pentium IV, 1.6 GHz CPU, 1Gb RAM pc. The algorithm was tested on instances with 5 to 20 customers and the results in computational time are shown in Figure 3.2.

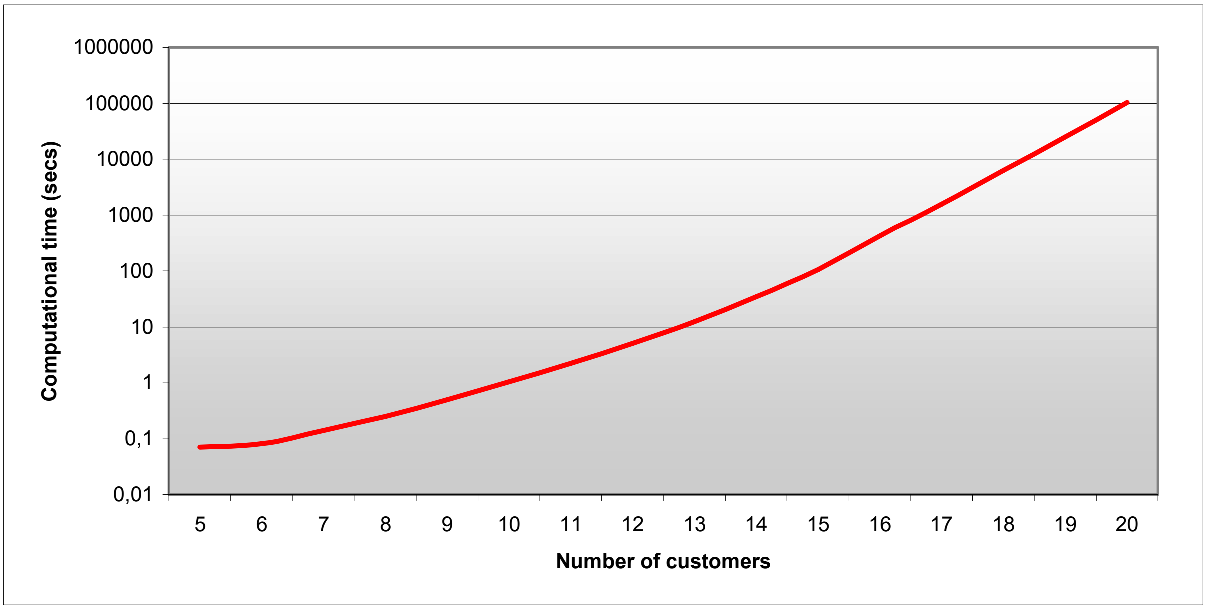


Figure 3.2. Computational time of VRDRP vs. Customer number.

From this figure it can be clearly seen that the computational time, and, therefore, the problem complexity, increases exponentially with the number of customers, as expected. Furthermore, beyond 20 customers the exhaustive search procedure is not practical.

3.3.2 Number of depot returns

An interesting characteristic of the problem is the number of times the vehicle will return to the depot to reload, in the optimal solution. The minimum number of depot returns is determined by finding the number of the absolute necessary returns without considering the distance traveled by the vehicle; i.e. the vehicle returns to the depot only when the stock on board is not sufficient to serve the next customer. This lower limit is defined as g_{min} and depends strictly on the customer demand vector ($d_1, d_2, \ldots, d_n$) and the vehicle capacity Q.

Let's now define g_{best} as the number of depot returns included in the best route of the vehicle. Obviously $g_{min} \leq g_{best}$. We investigated the relationship between g_{best} and g_{min} by creating 10,000 random problems and solving them to optimality, as follows:

1. Initiate the iterative procedure for 10,000 problems
2. For each problem create a random customer matrix i.e. random distances among customer points and depot, and random customer demands
3. Determine g_{min} by deriving the solution that is strictly based on vehicle capacity restrictions; that is the sum of customer demands between two consecutive depot visits must not exceed the vehicle capacity
5. Initialize the total distance traveled as Min = ∞
4. Do while $1 \leq i \leq n - g_{min}$
   a. Do while all possible combinations have not been exhausted:
      i. Create the first/next combination which contains g_{min+i} (i = 0,1,2,…n) returns to the depot (including the final return) as [1 0 0 …1]
      ii. Examine feasibility of this combination and if feasible
         1. Calculate total distance traveled (Dist)
         2. If Dist < Min then Min = Dist
         3. If Dist > Min then Min = Min
      iii. If the combination is not feasible go to step (i)
   b. End Do while
   c. $i = i + 1$
6. End Do while
7. Compare the best solutions identified for all g_{min+i} and identify the solution with the shortest distance
8. Record the shortest distance along with the number of depot returns of step 7
9. Repeat steps 2-7 for the next iteration
10. Conclude after 10,000 iterations have been completed.

The results of this experiment are shown in Figure 3.3.

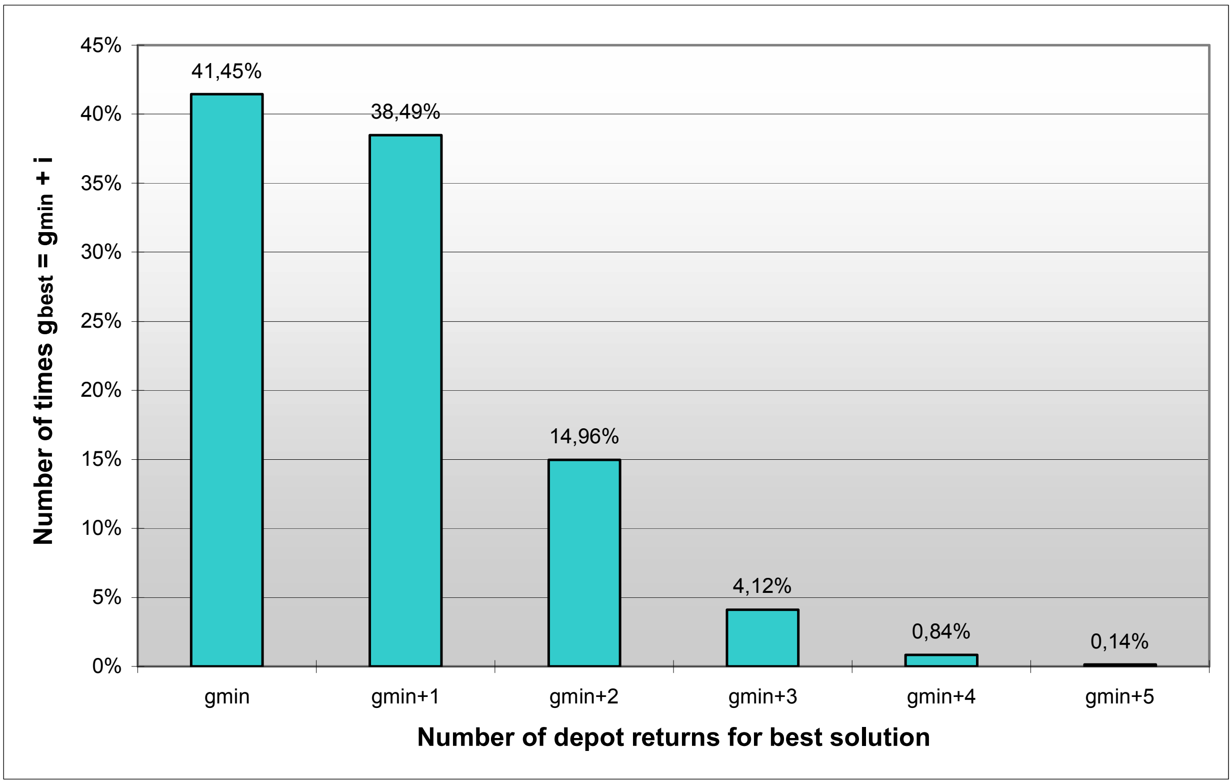


Figure 3.3. The results of the 10,000 experiments with 10 customers

According to these results it appears that the best solution so far contains g_{min} to $g_{min}+2$ depot returns 95 times out of 100 (95%). Therefore, in an enumerative algorithm that calculates all feasible solutions with g_{min}, $g_{min}+1$, and $g_{min}+2$ returns, the optimal solution would be determined with an approximately 95% confidence level.

3.4 An Efficient Solution: The Dynamic Programming Algorithm (DPA)

The VRDRP is solved to optimality using a dynamic programming algorithm. This is inspired by the work of Yang *et al.* (2000) and Manfrin *et al.* (2004), who presented the dynamic programming formulation for the Stochastic Vehicle Routing with Depot Returns Problem (VRDRP), where the demands of the customers are independent discrete random variables with known distributions.

Consider the example customer network presented in Figure 3.4.

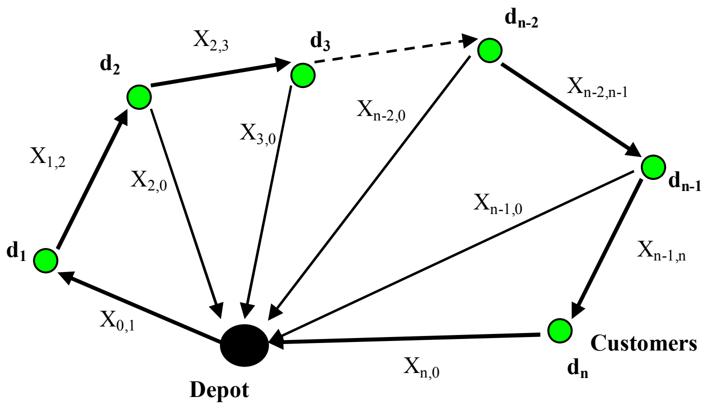


Figure 3.4. Small customer network.

Let $\mathbf{V_k(z)}$, k = n, n-1… 1 and z = 0… Q-d_k, be the minimum distance from customer $\boldsymbol{k}$, from which the vehicle departs with stock left on board equal to $\boldsymbol{z}$, to the end of the route.

For k= n:

$$V_n(z) = x_{n,0} \quad \text{and} \quad z = 0, 1\ldots Q-d_n \qquad (3.9)$$

For k= n-1:

$$V_{n-1}(z) = x_{n-1,n} + V_n(z) \qquad \text{if } z \geq d_n \qquad (3.10)$$

$$= x_{n-1,0} + x_{0,n} + V_n(z) \qquad \text{if } z < d_n \qquad (3.11)$$

For k= n-2…1:

$$V_k(z) = x_{k,0} + x_{0,k+1} + V_{k+1}(Q-d_{k+1}) \qquad \text{if } z < d_{k+1} \qquad (3.12)$$

$$= \min \{ x_{k,0} + x_{0,k+1} + V_{k+1}(Q-d_{k+1}), x_{k,k+1} + V_{k+1}(z-d_{k+1})\} \qquad \text{if } z \geq d_{k+1} \qquad (3.13)$$

Note that if $z < d_{k+1}$ the only feasible action is to return to the depot in order to refill and then go to customer $k+1$. If $z \geq d_{k+1}$ then there are two possible actions. If:

$$x_{k,k+1} + V_{k+1}(z-d_{k+1}) \leq x_{k,0} + x_{0,k+1} + V_{k+1}(Q-d_{k+1})$$

then the optimal decision is to go directly to customer $k+1$. If the reverse inequality holds, then the optimal decision is to return to the depot and then go to customer $k+1$. The total cost of the optimal route is equal to:

$$V_0(z) = x_{01} + V_1(Q-d_1) \qquad \text{for } k=0 \qquad (3.14)$$

The above dynamic programming algorithm determines the route with the minimum cost, following the reverse customer order. To do so, the minimum cost from each customer site to the end of the route is computed for all possible values of the vehicle load after the current customer has been served. Having completed these computations for all customers, the algorithm selects the arcs, which comprise the route with the overall minimum cost.

For the relevant theory of dynamic programming and some applications (e. g. knapsack problems, production and inventory models), in which a similar methodology is followed we point to Chapters 1-4 of Smith's (1991) book.

The exact steps of the algorithm are described below:

1 For a given customer matrix (distances and demands)
2 Start from the last customer and calculate $V_n(z)$ using Eq. (3.9)
3 Continue with the previous customer to calculate all $V_{n-1}(z)$ using Eq. (3.10) and Eq. (3.11)
4 Continue for all remaining customers (for k= n-2…1) to calculate all $V_k(z)$ using Eq. (3.12) and Eq. (3.13)
5 Compute the total cost of the optimal route using Eq.(3.14).
6 Determine the optimal route
7 End

The optimal route is determined by the values of the decision variables x_i that correspond to the values of $V_i(q)$ used for the computation of the minimum total cost. The algorithm identifies this value for each customer site and records the decision made. This decision can either be to proceed to the next customer site directly, or via the depot.

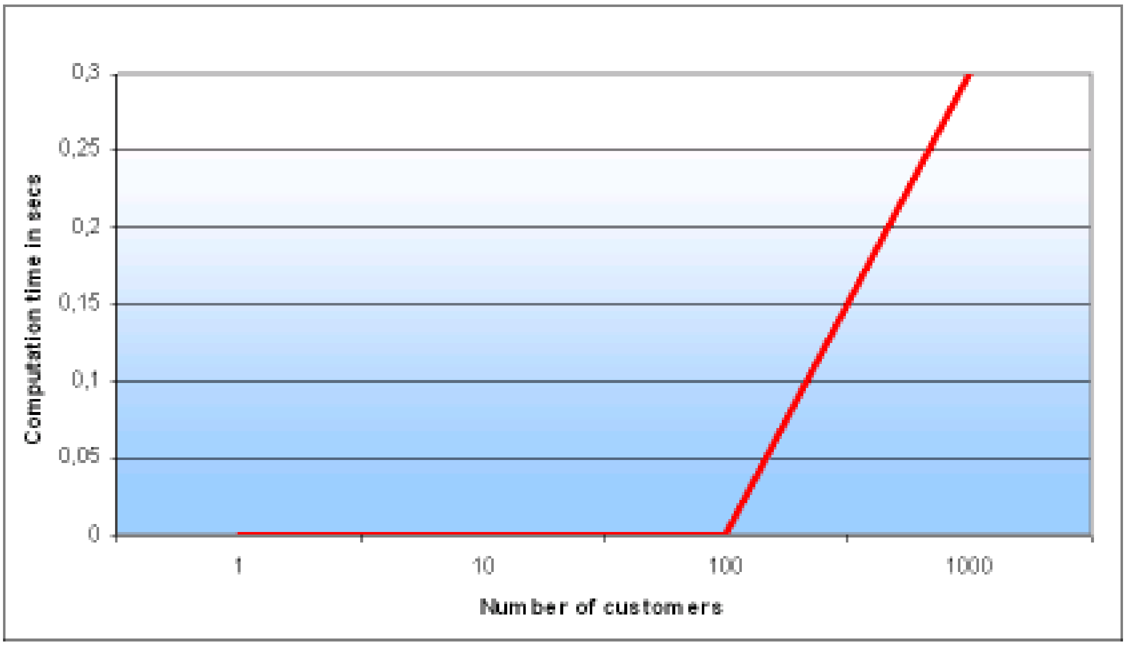


Figure 3.6. The results of the DPA Algorithm.

As it can be clearly observed in Figure 3.6, the performance of the Dynamic Programming Algorithm proved to be very efficient, as expected. For problem instances with up to 100 customers the algorithm obtained the results almost instantly. For 1000 customers the average calculation time is 0.3 seconds. The experiments were run on a PC equipped with Intel Pentium IV, at 2.4 GHz, and 512 MB of RAM.

3.5 Conclusions

In this chapter we presented the Vehicle Routing with Depot Returns Problem (VRDRP). The objective of the problem is to minimize cost (distance) while serving all customers in a predefined sequence with a single vehicle. The analysis of the problem showed that its complexity increases exponentially with the number of customers. Furthermore, the number of depot returns in the optimal solution g_{best} is most of the times between $g_{min} \leq g_{best} \leq g_{min+2}$ where g_{min} is the minimum feasible number of depot returns. Finally, a dynamic programming algorithm (DPA) inspired by the work of Yang *et al.* (2000) and Manfrin *et al.* (2004) was developed to solve the problem to optimality in efficient computational times.

Chapter 4

Variations of the VRDRP

4.1 Introduction

In Chapter 3 we saw that the optimal routing of a single vehicle, with limited capacity that delivers one product to n clients according to a predefined sequence, can be determined using dynamic programming. In this Chapter we propose and investigate two practical variations of this problem: (i) the case of multiple-product deliveries in which each product is stored in its own compartment in the vehicle and (ii) the case of multiple-product deliveries in which all products are stored together in the vehicle's single compartment. This work is the result of a joint effort with P.Tsiribas of the DeOPSys lab of the University of the Aegean. The problem formulations as well as the dynamic programming algorithms for the first variation were developed by the author of this dissertation. The algorithm for the second variation was the result of a joint effort with P.Tsiribas.

4.2 Variations of the VRDRP

The general problem setting for both variations is identical to that of the VRDRP; i.e. a single vehicle serves clients in a predefined sequence. We define below the particulars of the two problems.

Multiple-product Delivery - Compartmentalized load

We assume that the vehicle is divided into K sections and each section is suitable for one type of product only (see Figure 4.1a); i.e, section j is suitable for product $j \in \{1,\ldots,K\}$. A typical example of this situation is gasoline tankers. Let Q_j be the capacity of the vehicle for product j, for $j = 1,\ldots,K,$. Clearly, $\sum_{j=1}^{K} Q_j = Q$.

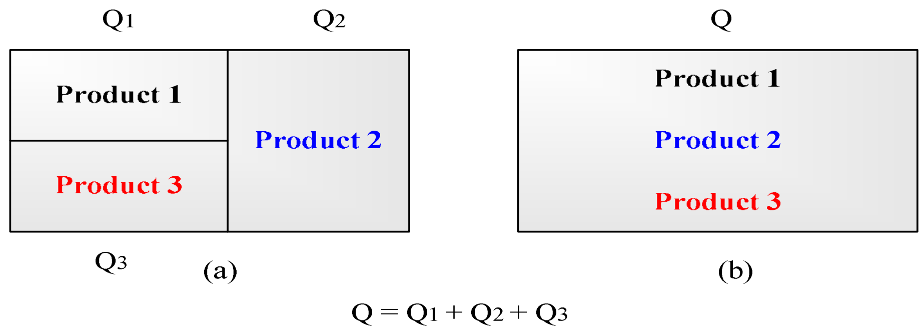


Figure 4.1. The multiple product extension.

Note that all product quantities are calculated using the same unit of measure e.g. m^3 or kg. We declare d_{ji} the demand of customer $i \in \{1,\ldots,n\}$ for product $j \in \{1,\ldots,K\}$. It is assumed that this demand cannot exceed the respective capacity of the vehicle, i.e. $d_{ji} \le Q_j$, for all $i = 1,\ldots,n$. The objective is to identify the nodes from which the vehicle will return to the depot for stock replenishment in order to minimize the total route cost. This problem will be addressed hereafter as Problem 1.

Multiple-product Delivery - Unified load

It differs from Problem 1 only in that the vehicle may carry any quantity for each product $j \in \{1, \ldots, K\}$, provided that the total capacity Q of the vehicle is not exceeded (see Figure 4.1b). This problem will be addressed hereafter as Problem 2.

4.3 The Solution algorithms

It is possible to design suitable dynamic programming algorithms for the two problems defined in the previous section. For each problem the dynamic programming algorithm determines the route with the minimum cost, following the reverse customer order. To do so, the minimum cost from each customer site to the end of the route is computed, for all possible values of the vehicle load after the current customer has been served. Having completed these computations for all customers, the algorithm selects the arcs, which compile the route with the overall minimum cost. We develop these algorithms below separately for each problem.

4.3.1 Algorithm for Problem 1

Let $V_i(z_1, \ldots, z_K), i = 1, \ldots, n,$ be the minimum total cost from customer *i* to the end of the route, if customer *i* has been served and the remaining quantity in the vehicle for product $j, 1 \le j \le K,$ is $z_j \in \{0, \ldots, Q_j - d_{ji}\}$. These quantities can be computed by using the following equations (4.1)-(4.5):

$$V_n(z_1, \ldots, z_K) = c_{n0}, \tag{4.1}$$

$$V_{n-1}(z_1, \ldots, z_K) = \begin{cases} c_{n-1,n} + V_n(z_1 - d_{1n}, \ldots, z_K - d_{Kn}) & \text{if} \quad \forall\, j \in \{1, \ldots, K\} : z_j \ge d_{jn}, \quad (4.2) \\ \\ c_{n-1,0} + c_{0n} + V_n(Q_1 - d_{1n}, \ldots, Q_K - d_{Kn}) & \text{if} \quad \exists\, j \in \{1, \ldots, K\} : z_j < d_{jn}. \quad (4.3) \end{cases}$$

For $i = n-2, \ldots, 1$:

$$V_i(z_1,\dots,z_K)=\begin{cases} c_{i0}+c_{0,i+1}+V_{i+1}(Q_1-d_{1,i+1},\dots,Q_K-d_{K,i+1}), \quad \text{if } \exists\, j\in\{1,\dots,K\}: z_j<d_{j,i+1}, & (4.4)\\[2ex] \min\{c_{i0}+c_{0,i+1}+V_{i+1}(Q_1-d_{1,i+1},\dots,Q_K-d_{K,i+1}), \\ c_{i,i+1}+V_{i+1}(z_1-d_{1,i+1},\dots,z_K-d_{K,i+1})\}, \quad \text{if } \forall\, j\in\{1,\dots,K\}: z_j\ge d_{j,i+1}. & (4.5)\end{cases}$$

Note that if there exists some $j\in\{1,\dots,K\}$ such that $z_j<d_{j,i+1}$, the only feasible action is to return to the depot in order to refill and then to go to customer $i+1$. If $z_j\ge d_{j,i+1}$ for all $j\in\{1,\dots,K\}$, then there are two possible actions. If

$$c_{i,i+1}+V_{i+1}(z_1-d_{1,i+1},\dots,z_K-d_{K,i+1})\le c_{i0}+c_{0,i+1}+V_{i+1}(Q_1-d_{1,i+1},\dots,Q_K-d_{K,i+1}),$$

then the optimal decision is to go directly to customer $i+1$. If the reverse inequality holds, then the optimal decision is to return to the depot and then to go to customer $i+1$. This case is possible, since due to the geometry of the route, a necessary return from a “remote” customer site may be avoided by returning to the depot from a customer site that is “close” to it and loading the vehicle up to its capacity. The total cost of the optimal route is equal to:

$$c_{01}+V_1(Q_1-d_{11},\dots,Q_K-d_{K1}). \tag{4.6}$$

Let $x_i(z_1,\dots,z_K)\in\{0,1\}$ represent the decision of the vehicle in node $i\in\{1,\dots,n\}$. Suppose that $x_i(z_1,\dots,z_K)=0$ when the vehicle goes to customer $i+1$ and $x_i(z_1,\dots,z_K)=1$ when it returns to the depot. It is clear that

$$x_n(z_1,\dots,z_K)=1,$$
$$x_i(z_1,\dots,z_K)=1, \text{ if } \exists\, j\in\{1,\dots,K\} \text{ such that } z_j<d_{j,i+1}, \text{ for } i\in\{1,\dots,n-1\},$$
$$x_i(z_1,\dots,z_K)\in\{0,1\}, \text{ if } \forall\, j\in\{1,\dots,k\}: z_j\ge d_{j,i+1}, \text{ for } i\in\{1,\dots,n-1\}.$$

The steps of the algorithm that determines the optimal policy for this problem are given below.

1. Compute and store $V_n(z_1,\ldots,z_K)$ for all acceptable values of $(z_1,\ldots,z_K)$ for node n, using (4.1); store the corresponding value of $x_n(z_1,\ldots,z_K)$.
2. Compute and store $V_{n-1}(z_1,\ldots,z_K)$ for all acceptable values of $(z_1,\ldots,z_K)$ for node $n-1$, using (4.2) and (4.3); store the corresponding values of $x_{n-1}(z_1,\ldots,z_n)$.
3. Compute and store for $i=n-2,\ldots,1$ the quantity $V_i(z_1,\ldots,z_K)$ for all acceptable values of $(z_1,\ldots,z_K)$ for node i, using (4.4) and (4.5); store the corresponding values of $x_i(z_1,\ldots,z_K)$.
4. Compute the total cost of the optimal route using (4.6). The optimal route is determined by the values of the decision variables x_i that correspond to the values of $V_i(z_1,\ldots,z_K)$ used for the computation of the minimum total cost.

Attention should be drawn to the fact that the algorithm described above is similar to the one presented in Section 3.4 of this dissertation.

4.3.2 Algorithm for Problem 2

This problem is transformed to the original VRDRP (i.e. Problem 1 with $K=1$), by computing the total customer demand (note that all product quantities are calculated using the same units of measure e.g. m^3 or kg). That way, it can be assumed that there is only one product type, and thus the problem is solved by implementing the algorithm for the original VRDRP problem. The proposed approach includes 3 steps:

1. Transform the problem to the original VRDRP (Problem 1 with $K=1$) The total demand d_i for customer i becomes

$$d_i = \sum_{j=1}^{K} d_{ji},\ i=1,\ldots,n.$$

2. Solve the VRDRP using the dynamic programming algorithm that we described in Problem 1 with $K=1$.

3. Suppose that according to the optimal route the vehicle returns to the depot after visiting the customers $i_1, \ldots, i_m$, where $m \le n$ and $i_1 < \cdots < i_m$. Then the quantity of product $j \in \{1, \ldots, K\}$ that the vehicle must load when it returns to the depot after serving customer $i_r, r \in \{1, \ldots, m\}$ must be equal to $\sum_{i=i_r+1}^{i_{r+1}} d_{ji}$.

In Step 3 we identify the stock quantity on board per product in the beginning of the route, as well as the quantity of refill per product, at each depot return. For each product, and for each sub-route (route between two subsequent depot returns) we add the demand of each customer site served in the particular sub-route. The sum of this total demand per product for the sub-route is equal to the optimal stock quantity that the vehicle should carry during this sub-route. The procedure is repeated for each sub-route until the entire route is exhausted and all product load quantities per sub-route are identified.

4.4 Implementation and Computational Analysis

In order to illustrate the proposed algorithms of Section 4.3, we present an example for each of these problems. In both examples the number of customers is equal to 5. Subsequently in both cases, a large number of problems are generated and solved in order to study the efficiency of the algorithms. The latter were implemented using *Matlab v. 7.0* and ran using a personal computer equipped with an Intel Pentium IV, 2.4 GHz processor and 512 MB of RAM.

4.4.1 Illustrative Examples - Problem 1

Consider the 5-customer network of Fig. 4.2. The vehicle capacity is $Q = 10$ units and is equally split between two products ($Q_1 = Q_2 = 5$); the demand for delivery $d_{k,i}$ for each product k (k = 1,2) and customer j (j = 1, …, 5), as well as the distances between the nodes c_{ij} are given in Fig. 4.2.

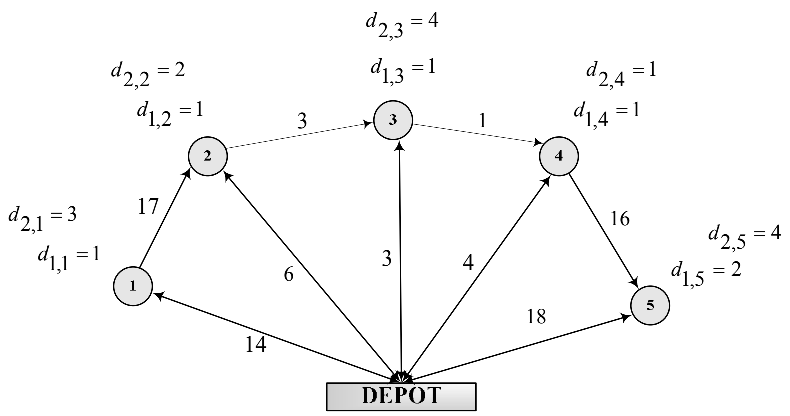


Figure 4.2. 5-customer network for the multiple-product (compartmentalized).

The problem is solved using the dynamic programming algorithm presented in section 4.3.1. Let $V_i(z_1, z_2)$ and $x_i(z_1, z_2)$ be the minimum total cost and the corresponding decision after customer $i \in \{1, 2, 3, 4, 5\}$ has been served. The remaining quantity in the vehicle is (z_1, z_2). Clearly, $V_5(z_1, z_2) = 18, x_5(z_1, z_2) = 1$ for $z_1 \in \{0, \ldots, 3\}$ & $z_2 \{0,1\}$. In Tables 4.1-4.4 we provide the results for nodes 1, 2, 3, 4. In these Tables, (z_1, z_2) represents the quantity carried by the vehicle after customer *i* has been served; each cell includes two values: The first is the value of $x_i(z_1, z_2)$ and the second is the value of $V_i(z_1, z_2)$.

Table 4.1. Results obtained for node 4

z1 / z2	0	1	2	3	4
0	1; 40	1; 40	1; 40	1; 40	1; 40
1	1; 40	1; 40	1; 40	1; 40	1; 40
2	1; 40	1; 40	1; 40	1; 40	1; 40
3	1; 40	1; 40	1; 40	1; 40	1; 40
4	1; 40	1; 40	0; 34	0; 34	0; 34

Table 4.2. Results obtained for node 3

z2 \ z1	0	1	2	3	4
0	1; 41	1; 41	1; 41	1; 41	1; 41
1	1; 41	1; 41	1; 41	1; 41	1; 41

Table 4.3. Results obtained for node 2

z2 \ z1	0	1	2	3	4
0	1; 50	1; 50	1; 50	1; 50	1; 50
1	1; 50	1; 50	1; 50	1; 50	1; 50
2	1; 50	1; 50	1; 50	1; 50	1; 50
3	1; 50	1; 50	1; 50	1; 50	1; 50

Table 4.4. Results obtained for node 1

z2 \ z1	4
2	0; 67

Therefore, the minimum total cost is equal to $14 + 67 = 81$ and the optimal route is $[0,1,1,0,1]$ (Fig. 4.3).

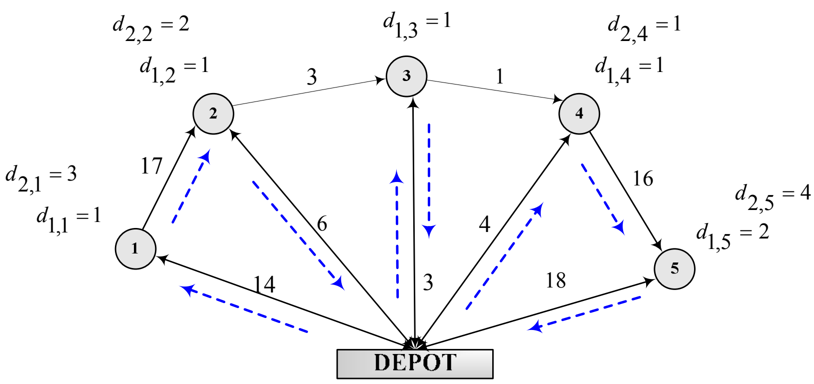


Figure 4.3. Optimal route for the 5-customer multiple-product problem.

4.4.2 Illustrative Examples - Problem 2

We consider the same example of Figure 4.2 for Problem 2. In this case however the vehicle is not compartmentalised. Following the steps of the algorithm of Section 4.3.2, first the problem is transformed to the original VRDRP presented in Section 3.2 (see Fig. 4.4).

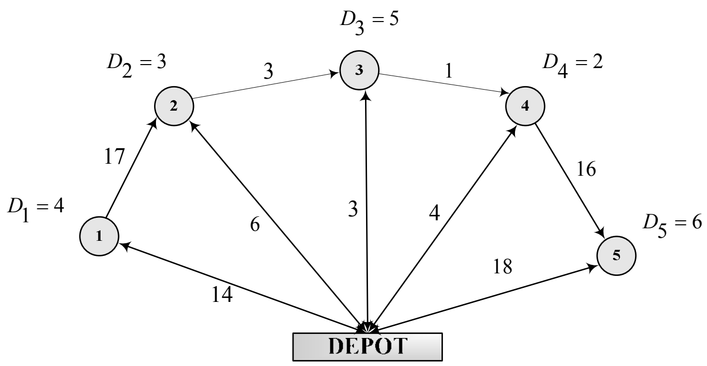


Figure 4.4. Transformation of the 5-customer multiple-product route.

According to Step 2 of the algorithm presented in Section 4.3.2, the basic problem is solved using the dynamic programming algorithm. Let $V_i(z)$ and $x_i(z)$ be the minimum total cost and the corresponding decision if customer $i \in \{1,2,3,4,5\}$ has been served and the remaining quantity in the vehicle is *z*. Clearly, $V_5(z) = 18, x_5(z) = 1$ for $z \in \{0,\ldots,14\}$. In Tables 4.5-4.8 we provide the results for nodes 1, 2, 3, 4. Again in these Tables, z represents the quantity carried to by the vehicle after customer *i* has been served; each cell includes two values: The first is the value of $x_i(z)$ and the second is the value of $V_i(z)$.

Table 4.5. Results obtained for node 4

z	0	1	2	3	4	5	6	7	8
	1; 40	1; 40	1; 40	1; 40	1; 40	1; 40	0; 34	0; 34	0; 34

Table 4.6. Results obtained for node 3

z	0	1	2	3	4	5
	1; 41	1; 41	1; 41	1; 41	1; 41	1; 41

Table 4.7. Results obtained for node 2

z	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	1; 50	1; 50	1; 50	1; 50	1; 50	0; 44	0; 44	0; 44

Table 4.8. Results obtained for node 1

z	**6**
	0; 64

The minimum total cost is equal to $14+64=78$ and the optimal route is $[1,0,1,0,1]$ (see Fig. 4.5).

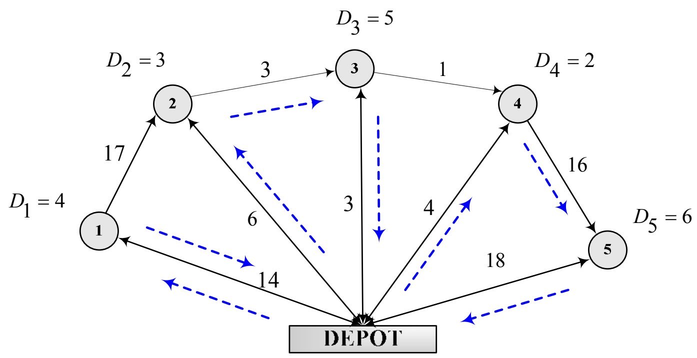


Figure 4.5. Optimal route for the 5-customer multiple-product problem.

In Table 4.9 we give, according to Step 3 of the algorithm, the quantities of the products 1, 2 that the vehicle must load when it leaves the depot a) at the beginning of the route and b) after serving customers 1 and 3. Individually for each product, and for each sub-route (route between two subsequent depot returns) we add the demand of each customer site. The sum of this total demand per product for the particular sub-route is equal to the optimal stock quantity that the vehicle should carry during this sub-route. The procedure is repeated for each sub-route until the entire route is covered and all optimal product load quantities per sub-route are identified.

Table 4.9. Loads for the 5-customer multiple-product problem

	Product 1	Product 2
Initial Load	1	3
Load after Customer 1	2	6
Load after Customer 4	3	5

Notice that when the vehicle leaves the depot a) at the beginning of the route and b) after serving customers 1 and 3, it is not loaded to its full capacity. This is due to the fact that according to the algorithm, the exact quantity of each product that the vehicle must carry for each sub-route is known. We therefore state that the algorithm can potentially create further cost savings, by preventing unnecessary loading / unloading of products at the depot.

According to the solution results, the minimum distance for the compartmentalized case was equal to 81 and the optimal route is $[0,1,1,0,1]$ (Fig. 4.3). Respectively, the minimum distance for the unified load case was equal to 78 and the optimal route is $[1,0,1,0,1]$ (Fig. 4.5). Both routes include 2 returns to the depot for refill in order to be able to fully satisfy the total demand. But the route of the unified load case makes the first depot return after serving Customer point-1 instead of after Customer point-2, therefore achieving a cost saving. This was possible due to the product quantity loading flexibility that the unified load case provides. It therefore becomes clear that it would be more cost effective to operate a fleet of unified load vehicles, as long as this is possible (according to the product types, if they can be stored at the same compartment, if they can be mixed, etc.).

4.5 Algorithm Performance Analysis

In this section we analyze the performance of the algorithms developed to solve each of the problems.

For the case of compartmentalized load (Problem 1) 3000 problems were created and solved. The number of customers in these problems ranged from 5 to 50, while the number of products was equal to 2, 3 and 4. The problems were formulated in the following manner: For each problem, a $n \times (3+K)$ matrix was created, where n denotes the number of customers (nodes) and K the number of product types. This matrix defines the network of the problem. Specifically, the first column relates to the customers, starting from customer *1* until customer n. The second column represents the distance of each customer from the depot, and the third the distances between customers according to the planned sequence of customer visits. The remaining columns represent the demand for each product type (K number of columns). The user provided the values of n and K, and the matrix values were generated randomly respecting the problem restrictions.

For the case of unified load (Problem 2) we generated 2000 problems following the procedure described above. In this case the number of customers ranged from 5 to 1000, while the number of products ranged from 2 to 5. All problems were solved using the algorithms presented in Section 4.3 and the computational times for each case (Problem 1 and Problem 2) are shown in Figures 4.6 and 4.7, respectively. From these results it is apparent that the complexity of the case of compartmentalized load is significantly higher than that of the case of unified load and so are the computational times.

Furthermore:

- The computational time for Problem 1 increases with a rate less than exponential with respect to the number of customers, while for Problem 2 it increases linearly.
- For Problem 1, the computational time increases exponentially with respect to the number of product types. This is due to the increase of the number of additional combinations to be examined for each additional product.

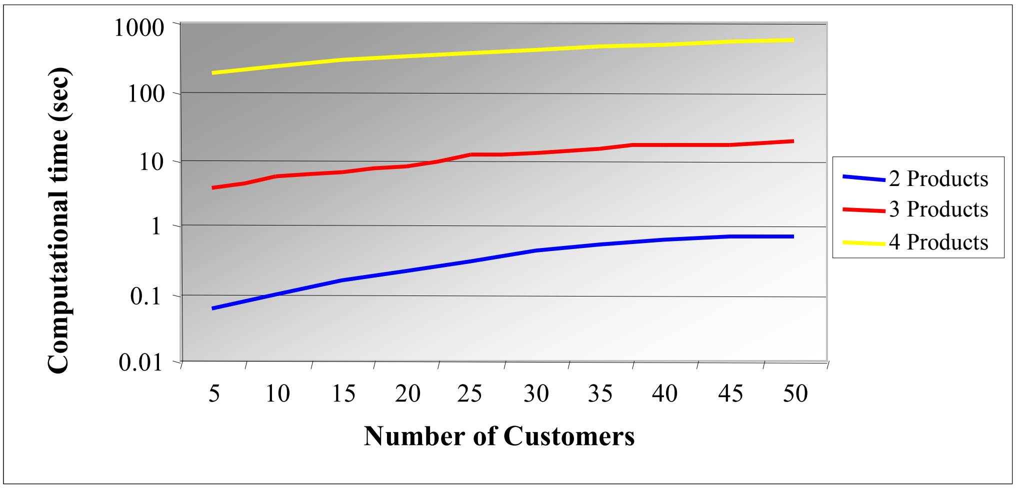


Figure 4.6. Computational time for Problem 1.

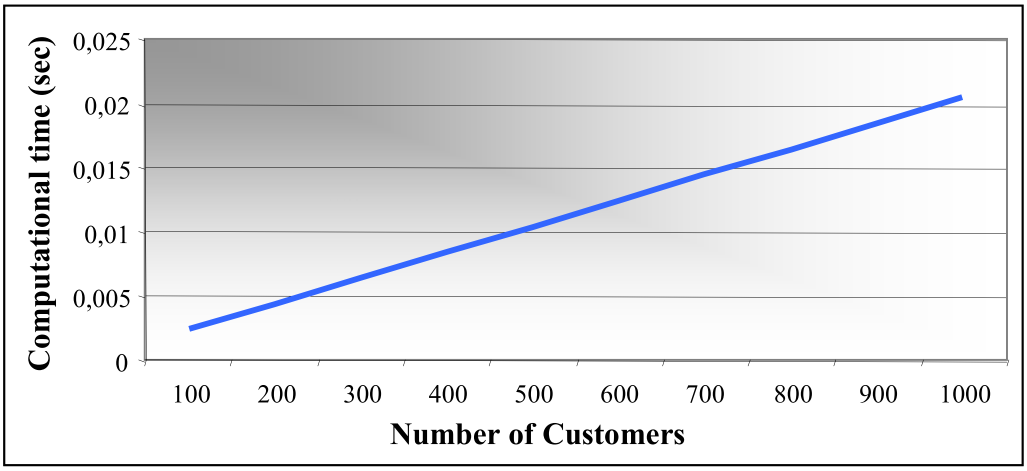


Figure 4.7. Computational time for Problem 2.

4.6 Conclusions

In this Chapter we presented two practical variations of the VRDRP: (i) The case of multiple-product deliveries when each product is stored in its own compartment in the vehicle and (ii) the case of multiple-product deliveries when all products are stored together in the vehicle's single compartment. The mathematical models, as well as efficient algorithms that solve the problems to optimality were developed and presented. Since these algorithms are optimal, neither further validation (e.g. Vs. Exhaustive Search Algorithm) nor comparison with other heuristic/metaheuristic methods is required. Both problems were approached by appropriate extensions of the dynamic programming algorithm presented in Section 3.4 of this dissertation. Based on the experimental results of the multiple-product problem, it has been demonstrated that the complexity of the compartmentalized case is significantly higher from that of the unified load case.

Chapter 5

The Stochastic Vehicle Routing with Depot Returns Problem (SVRDRP)

5.1 Introduction

In this Chapter we present and analyze the stochastic version of the Vehicle Routing with Depot Returns Problem (SVRDRP). In this problem the customer demands are independent random variables with known distributions. The SVRDRP has been initially presented and solved by Yang *et al.* (2000). The purpose of the chapter is to analyze the problem with respect to critical parameters that characterize the randomness of the demand. This lays the foundation for the cases presented in the following chapters.

5.2 The Stochastic Vehicle Routing with Depot Returns Problem (SVRDRP)

In the SVRDRP, the vehicle has been tasked to start from the depot, serve the customers at a predefined sequence and return to the depot. The demand of each customer is not known in advance, and it is revealed when the vehicle arrives at the customer site. Note that this situation is quite realistic. For example in the Ex-van case, the driver also acts as a sales person, negotiating with the customer the order quantities; thus these quantities can not be assumed to be known *a priori*. The customer demands are modeled by independent random variables with known statistics (derived from historical data). The distances, (or travel times), among all points in the network (depot and customer sites) are known. The product quantity loaded onto the vehicle is up to its capacity, which cannot be exceeded. The vehicle returns to the depot in order to refill as needed. It is assumed that service at a customer site (as long as there is enough stock to satisfy the demand in full), as well as refill at the depot occur instantly upon arrival of the vehicle at each location.

Upon completion of service at each customer, the driver has to make a decision: (a) Proceed to the next customer, as long as the probability of being able to fully satisfy the demand of the next customer is acceptable and the return distance is favourable; (b) return to the depot in order to refill, and resume the route by visiting the next customer in the predefined sequence. If the vehicle proceeds to the next customer but the actual demand of this customer turns out to be higher than the stock carried on board, the vehicle will unload its entire load, return to the depot to refill, and return to the customer to fully satisfy its demand (see Figure 5.1).

According to the example of Fig. 5.1, after serving customer 3 the vehicle's driver is faced with a decision: Either proceed directly to customer 4 following route (a), or via the depot following route (b). In case (a), if the actual demand of customer 4 turns out to be higher that the stock in the vehicle, then the customer's demand will be partially satisfied and a recourse action will be taken shown as route (c); i.e. the vehicle will return to the depot to refill, and proceed back to customer 4 to satisfy the remainder of this customer's demand.

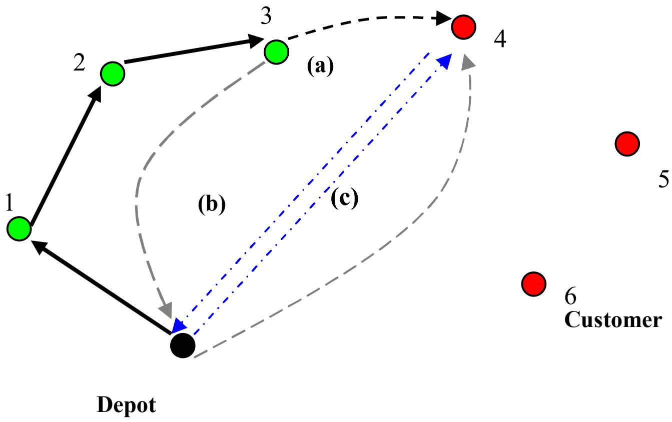


Figure 5.1. The decision by the vehicle's driver at each customer site.

To formalize the model of the SVRDRP, consider a set of nodes $V = \{0, \ldots, n\}$, with node 0 denoting the depot and nodes $1, \ldots, n$ corresponding to customers, and a set of arcs $A = \{(j, j+1), (j,0), (0, j+1) : j \in V - \{n\}\}$ that join the customers along the route $1 \to 2 \to \cdots \to n$, as well as all customers with the depot. The travel cost (distance) of each arc (i, j) is denoted by $c_{ij} > 0$. The c_{ij} values satisfy the triangular inequality. The network constraints of Eqs. (3.2) – (3.3) of Chapter 3 also hold here. It is assumed that the maximum capacity of the vehicle is equal to Q and the stock on board the vehicle upon service completion at customer j is equal to z. The random demand at customer j is given by ξ_j, where $\mathbf{P}(\xi_j = \xi^k)$ is known for every customer j and every integer $k \geq 0$. Also $\xi_j \geq 0$. Taking into consideration that the Vehicle Routing with Depot Returns Problem (VRDRP) is NP-hard, its Stochastic version (SVRDRP) is also NP-hard and significantly more complex (Kall, 1992).

According to Yang *et al.* (2000), in order to identify an optimal route for a single vehicle, it is first necessary to develop an efficient procedure to evaluate a particular route, that is, to find its expected cost under an optimal restocking policy. Upon service completion at customer j, let the vehicle carry a remaining load z, and let $f_j(z)$ denote the minimum expected value of the cost from customer j onward. If S_j represents the set of all possible loads that a vehicle can carry after service completion at customer j, then, $f_j(z)$ for $z \in S_j$ satisfies the following dynamic programming recursion (Yang *et al.* (2000)):

$$f_j(z) = \min \begin{cases} c_{j,j+1} + \sum_{k:\xi^k \le z} f_{j+1}(z-\xi^k)p_{j+1,k} + \sum_{k:\xi^k > z} [2c_{j+1,0} + f_{j+1}(z+Q-\xi^k)]p_{j+1,k} & \text{(part a)} \\ \\ c_{j,0} + c_{0,j+1} + \sum_{k=1}^{m} f_{j+1}(Q-\xi^k)p_{j+1,k} & \text{(part b)} \end{cases} \tag{5.1}$$

with the boundary condition:

$$f_n(z) = c_{n0} \qquad z \in S_n \tag{5.2}$$

In Eq. (5.1) part (a) in the minimization represents the expected cost of going directly to the next customer. If $\xi^k \le z$ the vehicle can fully satisfy the demand – this is represented by the first term of part (a). On the other hand if $\xi^k > z$ the vehicle cannot fully satisfy the demand of the customer, and has to execute a round trip to the depot for stock replenishment - this is represented by the second term of part (a). Part (b) of Eq. (5.1) represents the expected cost of the restocking action. Dynamic programming is used to recursively determine the optimal policy.

Note that:

- part (b) of Eq. (5.1), i.e. $f_j^{(2)}(z)$, is independent of z.
- Yang *et al.* (2000) have shown that part (a) of Eq. (5.1), i.e. $f_j^{(1)}(z)$, is a monotonically decreasing function; i.e.:

$$\text{if } z_1 > z_2 \qquad \text{then} \qquad f_j^{(1)}(z_1) \le f_j^{(1)}(z_2)$$

The formal proof is given in this reference and Appendix A. Based on these two facts Figure 5.2 shows the typical variation of $f_j^{(1)}(z)$ and $f_j^{(2)}(z)$ with respect to z. From this Figure it is clear that if z_j^* is the intersection point of $f_j^{(1)}(z)$ and $f_j^{(2)}(z)$ then

$$f_j(z) = \begin{cases} f_j^{(2)}(z) & \text{if} \quad z \le z_j^* \\ \\ f_j^{(1)}(z) & \text{if} \quad z > z_j^* \end{cases} \tag{5.3}$$

Thus, the quantity z_j^* represents a threshold below which, the vehicle should return to the depot and reload. If the vehicle's load after serving customer j is above this threshold, then the preferable action is to continue to the next customer directly. At each stage of the dynamic programming algorithm $f_j^{(2)}(z)$ of Eq. (5.1) is computed. Then $f_j^{(1)}(z)$ is computed in descending order of z until its value exceeds the value of $f_j^{(2)}(z)$. The last value of z for which $f_j^{(1)}(z) \leq f_j^{(2)}(z)$ is the threshold z_j^* for customer j.

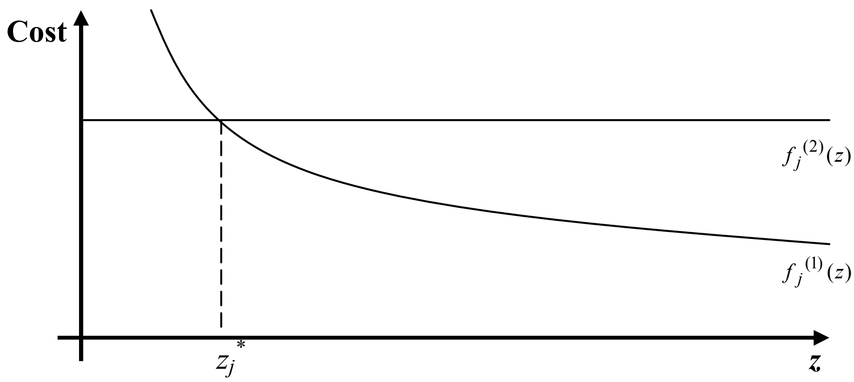


Figure 5.2. The typical variation of the two parts $f_j^{(1)}(z)$ and $f_j^{(2)}(z)$ of Eq. (5.1) with respect to z.

5.3 Analysis of the SVRDRP

In order to further explore the effect of randomness on the minimum expected cost, we analyzed the effect of the variance of the demand (demand randomness) on cost, as well as the effect of the interaction of the variance and the mean demand. Note that it is clear that increasing the mean demand will increase the expected cost of the route. This is because the capacity of the vehicle is constrained and more returns to the depot are necessary to satisfy the increased demand.

5.3.1 Effect of Randomness

In order to obtain a better insight on how the standard deviation of the random demands affects the cost of the route, the following experiment was designed as shown in Table 5.1 below. Column 1 of this Table identifies 5 cases of random demand for the 4 customers of Column 2. Columns 3-8 present the probability mass functions for z = 0,…,5 per customer per case. Columns 9, 10 present the average of the mean demand $\bar{\bar{x}}$ per customer and the variance s of the demand per case. Finally, Column 11 presents the result of the Yang *et al.* algorithm per case. Note that in all 5 cases the average mean demand per customer $\bar{\bar{x}}$ set remains constant while the variance increases.

Table 5.1. Randomness Analysis experiment.

Cases of Random Demand	Customers	Demand for item						Average Mean ($\bar{\bar{x}}$)	Yang et al Optimal solution	
		0	1	2	3	4	5		Variance	Minimum Expected Cost
1	1	0	0	1	0	0	0	2	$s = 0$	68
	2	0	1	0	0	0	0			
	3	0	0	0	1	0	0			
	4	0	0	1	0	0	0			
2	1	0	0,1	0,8	0,1	0	0	2	$s \approx 0{,}45$	72,36
	2	0,1	0,8	0,1	0	0	0			
	3	0	0	0,1	0,8	0,1	0			
	4	0	0,1	0,8	0,1	0	0			
3	1	0,1	0,1	0,6	0,1	0,1	0	2	$s \approx 1$	75,49
	2	0,2	0,6	0,1	0,1	0	0			
	3	0	0,1	0,1	0,6	0,1	0,1			
	4	0,1	0,1	0,6	0,1	0,1	0			
4	1	0,15	0,2	0,3	0,2	0,15	0	2	$s \approx 1{,}26$	77,87
	2	0,35	0,3	0,2	0,15	0	0			
	3	0	0,15	0,2	0,3	0,2	0,15			
	4	0,15	0,2	0,3	0,2	0,15	0			
5	1	0,2	0,2	0,2	0,2	0,2	0	2	$s \approx 1{,}41$	79,52
	2	0,4	0,2	0,2	0,2	0	0			
	3	0	0,2	0,2	0,2	0,2	0,2			
	4	0,2	0,2	0,2	0,2	0,2	0			

The solution with $s = 0$ has been obtained with the algorithm presented in Chapter 3. The solutions with $s > 0$ have been obtained using the Yang *et al.* algorithm presented in the current Chapter.

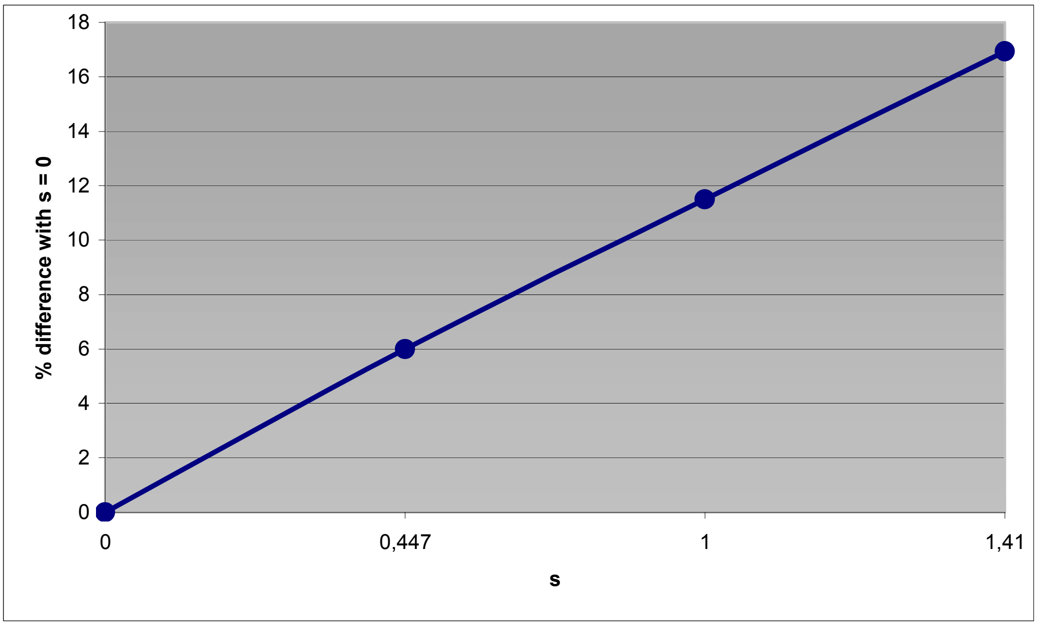


Figure 5.3. The relation between the value of s and the minimum expected cost of the route.

Figure 5.3 shows the percent increase of the minimum expected cost of the optimal solution, with respect to the deterministic case ($s = 0$) as a function of s. From this Figure it is clear that the expected cost of the route increases almost linearly with the variance of the demand. Thus, in the Ex-van business case, the consistency of Sales affects the distribution costs directly. Customers with inconsistent demand may lead to high distribution costs.

5.3.2 Mean – Variance Interaction

In this Section we investigate the interaction between the mean and the variance of the demand; that is, whether an increase of the mean will change the effect of the variance shown in Section 5.3.1. To analyze this we performed two additional experiments for the 4-customer case of Section 5.3.1 and for the cases of s = 0 and s = 1.4. For both new experiments the average mean demand is equal to 1 ($\bar{\bar{x}}$ = 1), lower than that of Table 5.1 ($\bar{\bar{x}}$ = 2). The data are shown in Table 5.2.

Table 5.2. Mean Analysis experiment.

Cases of Random Demand	Customers	Demand for item						Average Mean ($\bar{\bar{x}}$)	Yang et al Optimal solution	
		0	1	2	3	4	5		Variance	Minimum Expected Cost
1	1	0	1	0	0	0	0	1	(s = 0)	60
	2	1	0	0	0	0	0			
	3	0	0	1	0	0	0			
	4	0	1	0	0	0	0			
2	1	0,4	0,2	0,2	0,2	0	0	1	($s \approx$ 1,41)	67,21
	2	0,6	0,2	0,2	0	0	0			
	3	0,2	0,2	0,2	0,2	0,2	0			
	4	0,4	0,2	0,2	0,2	0	0			

The results are shown in Figure 5.4. From this Figure it is clear that there is an interaction between the mean and the variance of the demand; i.e. the percent increase of the minimum expected route cost for the low average mean demand case (11.6 %) is lower than the percent increase for the high average mean demand case (16.7%). That is, the randomness affects the expected cost more in vehicles with lower capacity.

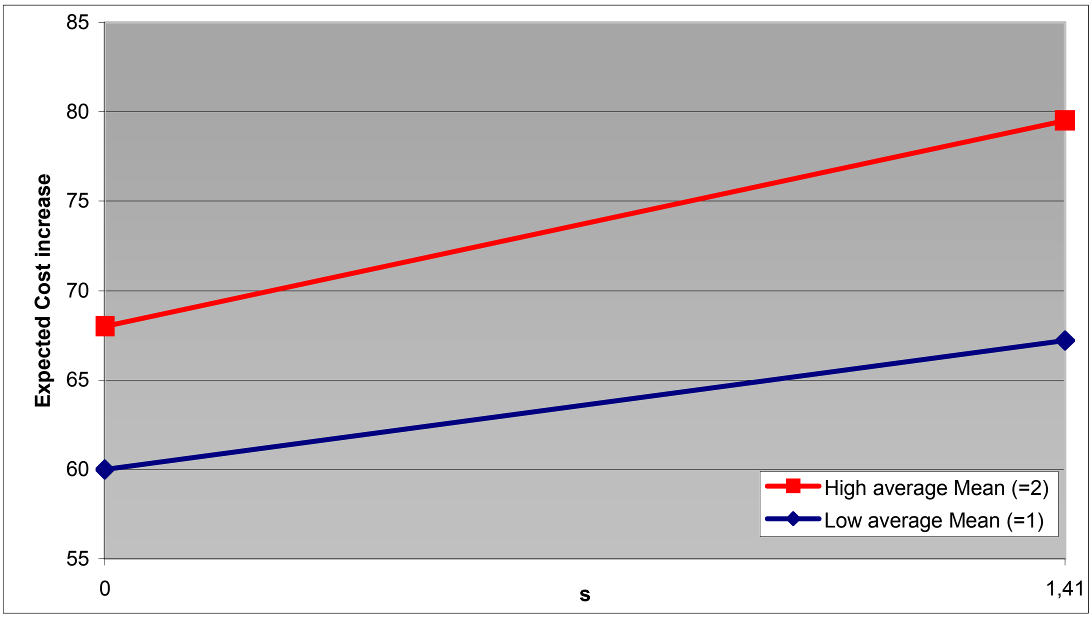


Figure 5.4. Mean-Variance interaction.

5.4 Conclusions

In this chapter we presented the Stochastic Vehicle Routing with Depot Returns Problem (SVRDRP). The objective of the problem is to minimize the expected travel cost (distance) while serving all customers in a predefined order with a single vehicle. First, the problem was analyzed to determine the effect of the variance of the demand on the minimum expected cost function. It was found that the expected cost of the route increases almost linearly with the variance of the demand. Thus, in the Ex-van business case, the consistency of Sales affects the distribution costs directly. Secondly, the problem was analyzed in order to determine the interaction between the mean and the variance of the demand. It was found that this interaction exists; i.e. the percent increase of the minimum expected route cost for the low average mean demand case (11.6 %) is lower than the percent increase for the high average mean demand case (16.7%) in the example. This interaction is reasonable, since the randomness affects the expected cost more in vehicles with lower capacity. Therefore, no further statistical analysis is warranted.

Chapter 6

Multiple Product Extensions of the SVRDRP

6.1 Introduction

In this Chapter we extend the Stochastic Vehicle Routing with Depot Returns Problem (SVRDRP) to address the case of distributing multiple product types. In line with Chapter 4 we address two cases; compartmentalized and unified load. The characteristics and the mathematical formulations of the two problems are presented. New algorithms are developed to solve both problems. The performance of these algorithms is analyzed by solving a large number of sample problems per case.

6.2 Multiple Product Delivery: Compartmentalized Case

In the business model addressed by this case, the vehicle carries different types of products, each within its own compartment. Thus, the capacity of the vehicle for each product is predefined and cannot be altered. A characteristic example is gasoline transport, in which the vehicle's tank is compartmentalized in order to carry various types of gasoline (unleaded, premium, etc.).

As before, the sequence of serving the customers is predefined. In this case, of course, the customer demand should be satisfied for all products. The demands per customer per product are independent discrete random variables, with known probability mass functions. The latter are derived in practice from historical data. The vehicle is allowed to return to the depot in order to refill. At the depot, all compartments may be refilled up to their capacity. Exactly upon completion of service at each customer site, the vehicle's driver has to make an identical decision with the one already discussed in Section 5.2, namely to continue towards the next customer or return to the depot.

The vehicle may have to visit a customer twice, if it cannot fully meet the demand of this customer during the first visit. It is assumed that service at a customer site and refill at the depot happen instantly upon arrival of the vehicle at the respective location. The objective of the problem is to serve all customers (replenish their stock of all items) and minimize the expected travel cost.

6.2.1 Dynamic Programming Formulation

We assume that the vehicle is divided into K sections and each section is suitable for carrying one product type only (see Figure 4.1a). Let Q_i be the capacity of the vehicle for product $i \in \{1,\dots,K\}$. Clearly, $\sum_{i=1}^{K} Q_i = Q$. Note that all product quantities are calculated using the same unit of measure e.g. m^3 or kg. Let z_i represent the stock on board of each product after serving customer j.

We declare ξ_j^i the stochastic demand of customer $j \in \{1,\dots,n\}$ for product type $i \in \{1,\dots,K\}$. ξ_j^i follows a discrete distribution with m_i possible values, ξ^{1_i}, ξ^{2_i}, …, ξ^{m_i} and probability mass function:

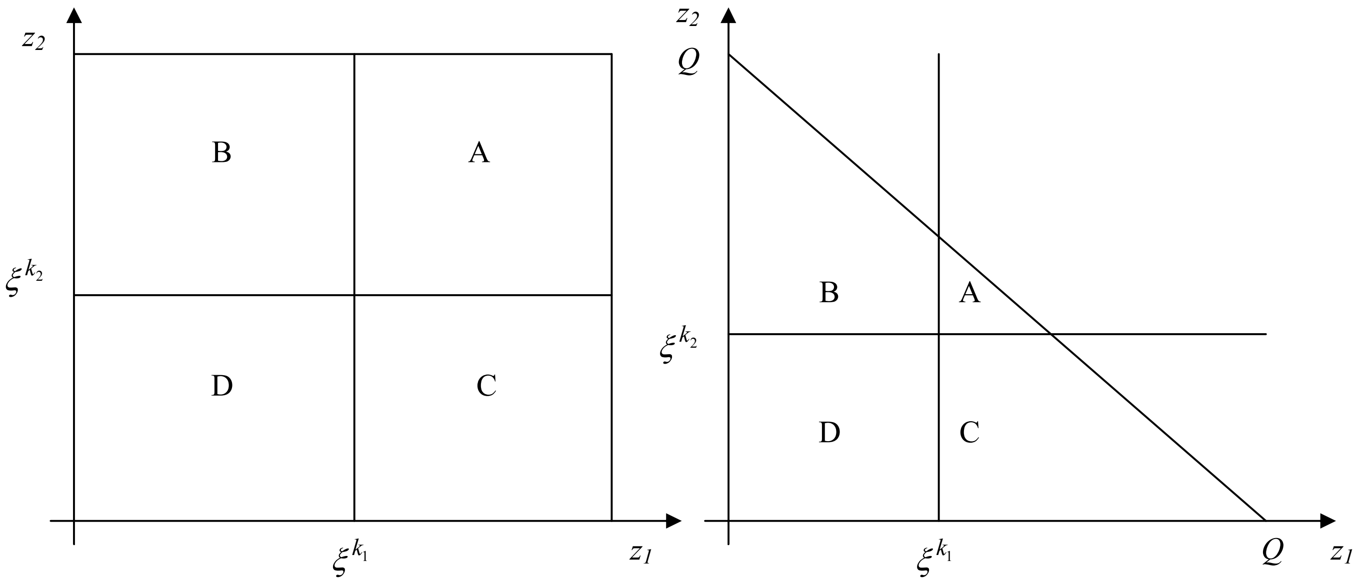
$$p^{i}_{jk} = \mathrm{P}(\xi^{i}_{j} = \xi^{k}) \tag{6.1}$$

For simplicity, but without loss of generality, we will develop the dynamic programming formulation for two product types (the formulation for 3 product types is presented in Appendix B).

Let $f_j(z_1, z_2)$ denote the total minimum expected cost from customer j onward if the vehicle, after serving customer j, carries quantities z_1 of product type-1 and z_2 of product type-2. If S_j represents the set of all possible loads that a vehicle can carry after serving customer j, then $f_j(z_1, z_2)$ for $(z_1, z_2) \in S_j$ satisfies the dynamic programming recursion:

$$f_j(z_1,z_2) = \min \begin{cases} \left.\begin{aligned} & c_{j,j+1} + \sum_{k_1:\xi^{k_1} \le z_1} \sum_{k_2:\xi^{k_2} \le z_2} f_{j+1}(z_1 - \xi^{k_1}, z_2 - \xi^{k_2}) p^1_{j+1,k_1} p^2_{j+1,k_2} + \\ & + \sum_{k_1:\xi^{k_1} > z_1} \sum_{k_2:\xi^{k_2} \le z_2} \left[2c_{j+1,0} + f_{j+1}(z_1 + Q_1 - \xi^{k_1}, Q_2)\right] p^1_{j+1,k_1} p^2_{j+1,k_2} + \\ & + \sum_{k_1:\xi^{k_1} \le z_1} \sum_{k_2:\xi^{k_2} > z_2} \left[2c_{j+1,0} + f_{j+1}(Q_1, z_2 + Q_2 - \xi^{k_2})\right] p^1_{j+1,k_1} p^2_{j+1,k_2} + \\ & + \sum_{k_1:\xi^{k_1} > z_1} \sum_{k_2:\xi^{k_2} > z_2} \left[2c_{j+1,0} + f_{j+1}(z_1 + Q_1 - \xi^{k_1}, z_2 + Q_2 - \xi^{k_2})\right] p^1_{j+1,k_1} p^2_{j+1,k_2} \end{aligned}\right\} \text{part (a)} \\ \left.c_{j,0} + c_{0,j+1} + \sum_{k_1=1}^{m_1} \sum_{k_2=1}^{m_2} f_{j+1}(Q_1 - \xi^{k_1}, Q_2 - \xi^{k_2}) p^1_{j+1,k_1} p^2_{j+1,k_2} \right\} \text{part (b)} \end{cases} \tag{6.2}$$

In Eq. (6.2), part (a) represents the expected cost of going directly to the next customer, whereas part (b) represents the expected cost of the restocking action. Part (a) consists of four summation terms (the four rows), which correspond to the four cases of Figure 6.1.



Part (a) - Compartmentalized Part (b) – Unified load

Figure 6.1. The solution space per customer point (for (a) compartmentalized and (b) unified load).

The first sum of part (a) represents the expected cost incurred if the stock of both items z_1 and z_2 is sufficient to fully satisfy the demands ξ^{k_1} and ξ^{k_2} of customer *j+1* (Area A in part (a) of Figure 6.1). The loads of the vehicle before and after serving customer *j+1* are schematically represented above the respective arcs (the demand of customer *j+1* is presented above the node) in Figure 6.2. The stock left on board after serving customer *j+1* is $(z_1 - \xi^{k_1}) \geq 0$ and $(z_2 - \xi^{k_2}) \geq 0$ respectively.

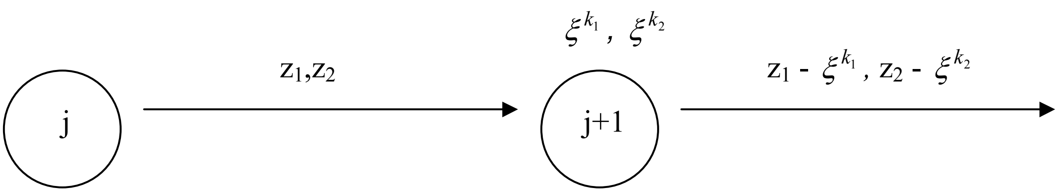


Figure 6.2. The case that the demand is fully satisfied: $z_1 \geq \xi^{k_1}$, $z_2 \geq \xi^{k_2}$.

The second sum of part (a) represents the expected cost incurred if the vehicle proceeds to the next customer directly and the stock z_1 is not sufficient to fully satisfy the demand ξ^{k_1} of the next customer, while the stock z_2 is sufficient to fully satisfy the demand ξ^{k_2} (Area B in part (a) of Figure 6.1). The path of the vehicle and the corresponding vehicle loads are shown in Figure 6.3.

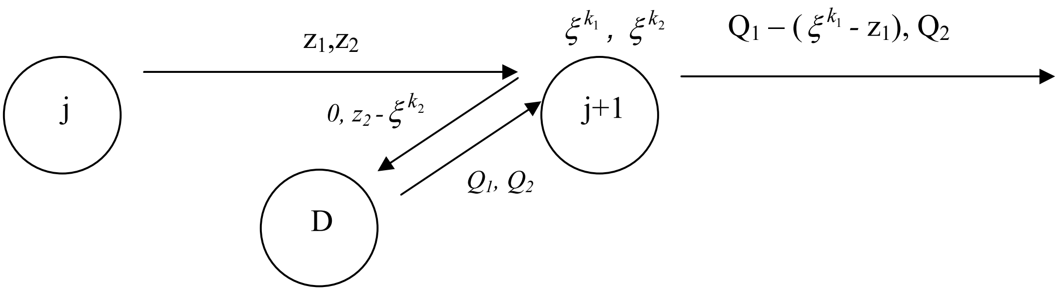


Figure 6.3. The case in which $z_1 < \xi^{k_1}$ and $z_2 \geq \xi^{k_2}$.

In this case, the vehicle will visit customer $j+1$, it will fully satisfy demand ξ^{k_2} for product type-2 but will partially satisfy demand ξ^{k_1} for product type-1; thus, it is required to return to the depot for stock replenishment. The stock left on board after serving customer $j+1$ for the first time is 0 and $(z_2 - \xi^{k_2}) \geq 0$, respectively. The stock left on board after serving customer j+1 is $Q_1 - (\xi^{k_1} - z_1)$ for product type-1 and Q_2 for product type-2 respectively.

The third sum of part (a) represents the expected cost incurred if the stock z_2 is not sufficient to fully satisfy the demand ξ^{k_2} of the next customer, while the stock z_1 is sufficient to satisfy demand ξ^{k_1} (Area C in part (a) of Figure 6.1). The path of the vehicle and the corresponding vehicle loads are shown in Figure 6.4 below. The load balance is analogous to the previous case.

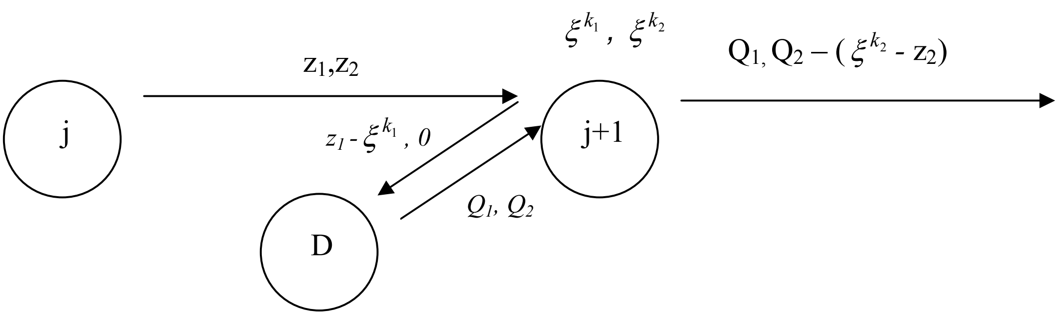


Figure 6.4. The case in which $z_2 < \xi^{k_2}$ and $z_1 \geq \xi^{k_1}$.

The fourth term represents the expected cost incurred if the stock of both items z_1 and z_2 is not sufficient to fully satisfy the demands ξ^{k_1} or ξ^{k_2} of the next customer (Area D in part (a) of Figure 6.1). The path of the vehicle and the corresponding vehicle loads are shown in Figure 6.5 below.

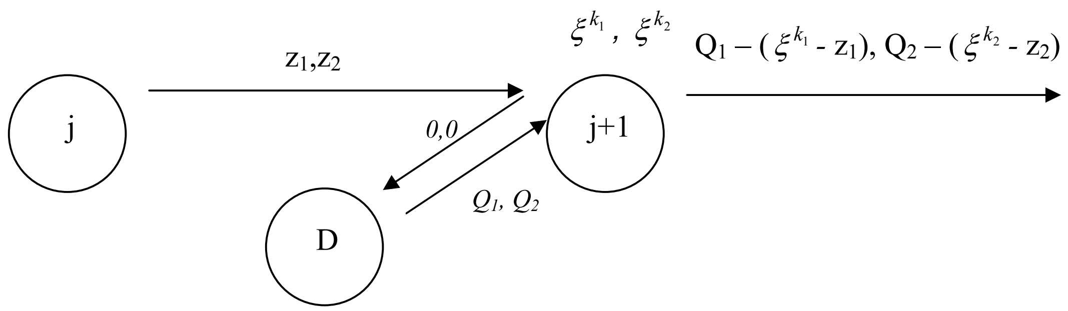


Figure 6.5. The case in both $z_1 < \xi^{k_1}$ and $z_2 < \xi^{k_2}$.

Part (b) of the minimization equation, Eq. (6.2), represents the expected cost incurred if the vehicle proceeds to the next customer $j+1$ via the depot and, therefore, the stock of both items on board z_1 and z_2 is replenished to Q_1 and Q_2 respectively. Due to the fact that this is a proactive depot return, the values of z_1 and z_2 do not affect the result, which is thus independent of z_1 and z_2. The path of the vehicle and the corresponding vehicle loads are shown in Figure 6.6 below.

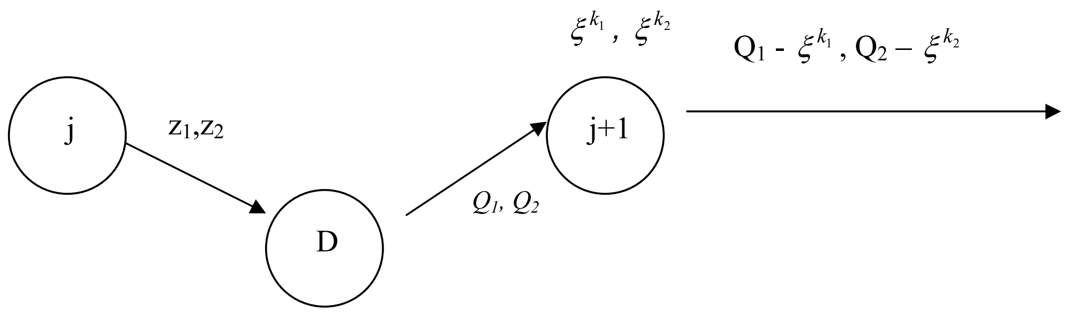


Figure 6.6. The proactive depot return case.

6.2.2 Optimal Routing Policy

The expected value of the minimum expected cost can be estimated using Eq. (6.2). However, in order to develop the policy which leads to achieving this minimum expected cost we will extend the threshold theorem presented by Yang *et al.* (2000) to multiple dimensions.

LEMMA 1.

$$f_j(z_1, z_2) \le f_j(Q_1, Q_2) + 2c_{0j} \quad \text{for all } z_1, z_2 \in S_j \tag{6.3}$$

Proof. From Eq. (6.2) we obtain:

$$f_j(z_1, z_2) \le c_{j,0} + c_{0,j+1} + \sum_{k_1=1}^{m_1} \sum_{k_2=1}^{m_2} f_{j+1}(Q_1 - \xi^{k_1}, Q_2 - \xi^{k_2}) p^1_{j+1,k_1} p^2_{j+1,k_2} \tag{6.4}$$

For $z_1 = Q_1$ and $z_2 = Q_2$ part (a) of Eq. (6.2) is always less than or equal to part (b) since $c_{j,j+1} \le c_{j,0} + c_{0,j+1}$; therefore:

$$f_j(Q_1, Q_2) = c_{j,j+1} + \sum_{k_1=1}^{m_1} \sum_{k_2=1}^{m_2} f_{j+1}(Q_1 - \xi^{k_1}, Q_2 - \xi^{k_2}) p^1_{j+1,k_1} p^2_{j+1,k_2} \tag{6.5}$$

taking into consideration that the last three terms of part (a) of Eq.(6.2) become zero.

Substituting Eq. (6.5) in Eq. (6.4) results in:

$$f_j(z_1, z_2) \le c_{j,0} + c_{0,j+1} + [f_j(Q_1, Q_2) - c_{j,j+1}]$$

$$\le c_{j,0} + [c_{0,j} + c_{j,j+1}] + [f_j(Q_1, Q_2) - c_{j,j+1}] \le 2c_{j,0} + f_j(Q_1, Q_2) \qquad \text{QED}$$

Consider z_1^*, z_2^* the item quantities on board after serving customer *j*.

THEOREM 1: *For each customer j, there exists a threshold function $h_j(z_1^*, z_2^*) = c_j$, such that the optimal decision, after serving customer j is to continue to customer j+1 if $h_j(z_1, z_2) \ge c_j$ or return to the depot otherwise.*

Proof: We will first show by induction that for all $(z_1,z_2) \in S_j$, $f_j(z_1,z_2)$ is a non-increasing function. That is, for $z_1, z_2 \in S_j$ and $\delta_1, \delta_2 \geq 0$

$$f_j(z_1+\delta_1, z_2+\delta_2) \leq f_j(z_1, z_2) \tag{6.6}$$

This relationship is true for the last customer n, where $f_n(z_1,z_2) = c_{n0}$ is independent of (z_1,z_2). Hence $f_n(z_1,z_2)$ is monotonically non-increasing with respect to $(z_1,z_2) \in S_n$. We will now prove that, if $f_{j+1}(z_1,z_2)$ is monotonically non-increasing with respect to $(z_1,z_2) \in S_{j+1}$, then $f_j(z_1,z_2)$ is also monotonically non-increasing with respect to $(z_1,z_2) \in S_j$. Let $H_j(z_1,z_2)$ and $H'_j(z_1,z_2)$ denote the values of part (a) and part (b) inside the minimisation in Eq.(6.2).

Let:

$$H_j(z_1, z_2) = H_j^a(z_1,z_2) + H_j^b(z_1,z_2) + H_j^c(z_1,z_2) + H_j^d(z_1,z_2) \tag{6.7}$$

where:

$$H_j^a(z_1,z_2) = c_{j,j+1} + \sum_{k_1:\xi^{k_1} \leq z_1} \sum_{k_2:\xi^{k_2} \leq z_2} f_{j+1}(z_1 - \xi^{k_1}, z_2 - \xi^{k_2}) p^1_{j+1,k_1} p^2_{j+1,k_2} \tag{6.7.1}$$

$$H_j^b(z_1,z_2) = \sum_{k_1:\xi^{k_1} > z_1} \sum_{k_2:\xi^{k_2} \leq z_2} \left[2c_{j+1,0} + f_{j+1}(z_1 + Q_1 - \xi^{k_1}, Q_2)\right] p^1_{j+1,k_1} p^2_{j+1,k_2}$$

$$H_j^c(z_1,z_2) = \sum_{k_1:\xi^{k_1} \leq z_1} \sum_{k_2:\xi^{k_2} > z_2} \left[2c_{j+1,0} + f_{j+1}(Q_1, z_2 + Q_2 - \xi^{k_2})\right] p^1_{j+1,k_1} p^2_{j+1,k_2}$$

$$H_j^d(z_1,z_2) = \sum_{k_1:\xi^{k_1} > z_1} \sum_{k_2:\xi^{k_2} > z_2} \left[2c_{j+1,0} + f_{j+1}(z_1 + Q_1 - \xi^{k_1}, z_2 + Q_2 - \xi^{k_2})\right] p^1_{j+1,k_1} p^2_{j+1,k_2}$$

If we expand each one of the last three terms taking into account the regions of Figure 6.7 we obtain:

$$\begin{aligned} H_j^b(z_1,z_2) = & \sum_{k_1:z_1<\xi^{k_1} \leq z_1+\delta_1} \sum_{k_2:\xi^{k_2} \leq z_2} \left[2c_{j+1,0} + f_{j+1}(z_1 + Q_1 - \xi^{k_1}, Q_2)\right] p^1_{j+1,k_1} p^2_{j+1,k_2} \\ & + \sum_{k_1:z_1+\delta_1<\xi^{k_1}} \sum_{k_2:\xi^{k_2} \leq z_2} \left[2c_{j+1,0} + f_{j+1}(z_1 + Q_1 - \xi^{k_1}, Q_2)\right] p^1_{j+1,k_1} p^2_{j+1,k_2} \end{aligned} \tag{6.7.2}$$

$$H_j^c(z_1,z_2) = \sum_{k_1:\xi^{k_1}\le z_1}\sum_{k_2:z_2<\xi^{k_2}\le z_2+\delta_2}\left[2c_{j+1,0} + f_{j+1}(Q_1, z_2+Q_2-\xi^{k_2})\right]p^1_{j+1,k_1}p^2_{j+1,k_2}$$
$$+ \sum_{k_1:\xi^{k_1}\le z_1}\sum_{k_2:z_2+\delta_2<\xi^{k_2}}\left[2c_{j+1,0} + f_{j+1}(Q_1, z_2+Q_2-\xi^{k_2})\right]p^1_{j+1,k_1}p^2_{j+1,k_2} \quad (6.7.3)$$

$$H_j^d(z_1,z_2) = \sum_{k_1:z_1<\xi^{k_1}\le z_1+\delta_1}\sum_{k_2:z_2<\xi^{k_2}\le z_2+\delta_2}\left[2c_{j+1,0} + f_{j+1}(z_1+Q_1-\xi^{k_1}, z_2+Q_2-\xi^{k_2})\right]p^1_{j+1,k_1}p^2_{j+1,k_2}$$
$$+ \sum_{k_1:z_1+\delta_1<\xi^{k_1}}\sum_{k_2:z_2<\xi^{k_2}\le z_2+\delta_2}\left[2c_{j+1,0} + f_{j+1}(z_1+Q_1-\xi^{k_1}, z_2+Q_2-\xi^{k_2})\right]p^1_{j+1,k_1}p^2_{j+1,k_2}$$
$$+ \sum_{k_1:z_1<\xi^{k_1}\le z_1+\delta_1}\sum_{k_2:z_2+\delta_2<\xi^{k_2}}\left[2c_{j+1,0} + f_{j+1}(z_1+Q_1-\xi^{k_1}, z_2+Q_2-\xi^{k_2})\right]p^1_{j+1,k_1}p^2_{j+1,k_2}$$
$$+ \sum_{k_1:z_1+\delta_1<\xi^{k_1}}\sum_{k_2:z_2+\delta_2<\xi^{k_2}}\left[2c_{j+1,0} + f_{j+1}(z_1+Q_1-\xi^{k_1}, z_2+Q_2-\xi^{k_2})\right]p^1_{j+1,k_1}p^2_{j+1,k_2} \quad (6.7.4)$$

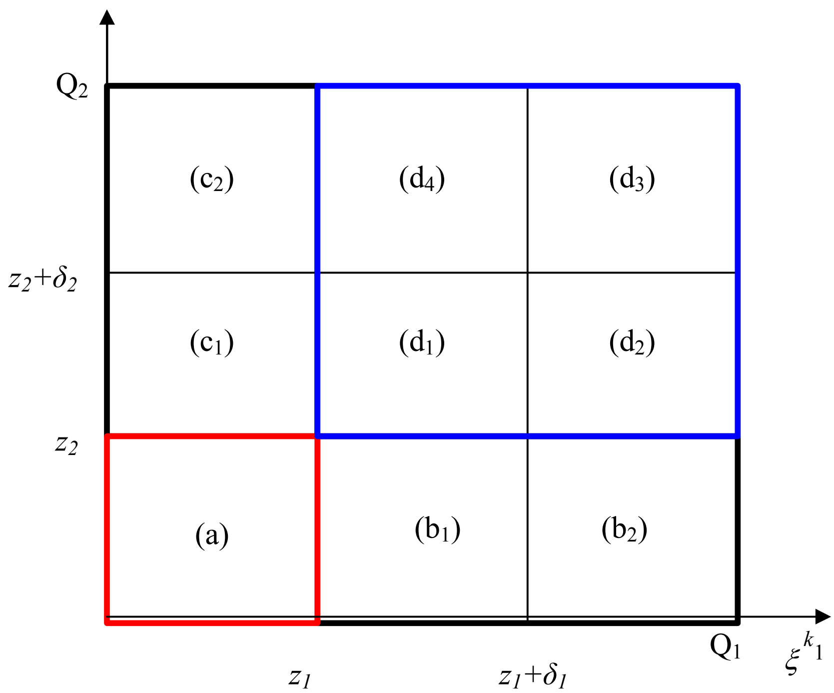


Figure 6.7. The definition space of $H_j(z_1, z_2)$.

Similarly $H_j(z_1+\delta_1, z_2+\delta_2)$ can be written using Figure 6.8 as follows:

$$H_j(z_1+\delta_1, z_2+\delta_2) = H_j^{a'}(z_1+\delta_1, z_2+\delta_2) + H_j^{b'}(z_1+\delta_1, z_2+\delta_2) + H_j^{c'}(z_1+\delta_1, z_2+\delta_2) + H_j^{d'}(z_1+\delta_1, z_2+\delta_2) \tag{6.8}$$

If we expand each one of the above terms taking into account the regions of Figure 6.8 we obtain:

$$\begin{aligned} H_j^{a'}(z_1+\delta_1, z_2+\delta_2) = c_{j,j+1} &+ \sum_{k_1:\xi^{k_1}\le z_1} \sum_{k_2:\xi^{k_2}\le z_2} f_{j+1}(z_1+\delta_1-\xi^{k_1}, z_2+\delta_2-\xi^{k_2}) p^1_{j+1,k_1} p^2_{j+1,k_2} \\ &+ \sum_{k_1:z_1<\xi^{k_1}\le z_1+\delta_1} \sum_{k_2:\xi^{k_2}\le z_2} f_{j+1}(z_1+\delta_1-\xi^{k_1}, z_2+\delta_2-\xi^{k_2}) p^1_{j+1,k_1} p^2_{j+1,k_2} \\ &+ \sum_{k_1:\xi^{k_1}\le z_1} \sum_{k_2:z_2<\xi^{k_2}\le z_2+\delta_2} f_{j+1}(z_1+\delta_1-\xi^{k_1}, z_2+\delta_2-\xi^{k_2}) p^1_{j+1,k_1} p^2_{j+1,k_2} \\ &+ \sum_{k_1:z_1<\xi^{k_1}\le z_1+\delta_1} \sum_{k_2:z_2<\xi^{k_2}\le z_2+\delta_2} f_{j+1}(z_1+\delta_1-\xi^{k_1}, z_2+\delta_2-\xi^{k_2}) p^1_{j+1,k_1} p^2_{j+1,k_2} \end{aligned} \tag{6.8.1}$$

$$\begin{aligned} H_j^{b'}(z_1+\delta_1, z_2+\delta_2) = &\sum_{k_1:\xi^{k_1}> z_1+\delta_1} \sum_{k_2:\xi^{k_2}\le z_2} \left[2c_{j+1,0} + f_{j+1}(z_1+\delta_1+Q_1-\xi^{k_1}, Q_2)\right] p^1_{j+1,k_1} p^2_{j+1,k_2} \\ &+ \sum_{k_1:\xi^{k_1}> z_1+\delta_1} \sum_{k_2:z_2<\xi^{k_2}\le z_2+\delta_2} \left[2c_{j+1,0} + f_{j+1}(z_1+\delta_1+Q_1-\xi^{k_1}, Q_2)\right] p^1_{j+1,k_1} p^2_{j+1,k_2} \end{aligned} \tag{6.8.2}$$

$$\begin{aligned} H_j^{c'}(z_1+\delta_1, z_2+\delta_2) = &\sum_{k_1:\xi^{k_1}\le z_1} \sum_{k_2:\xi^{k_2}> z_2+\delta_2} \left[2c_{j+1,0} + f_{j+1}(Q_1, z_2+\delta_2+Q_2-\xi^{k_2})\right] p^1_{j+1,k_1} p^2_{j+1,k_2} \\ &+ \sum_{k_1:z_1<\xi^{k_1}\le z_1+\delta_1} \sum_{k_2:\xi^{k_2}> z_2+\delta_2} \left[2c_{j+1,0} + f_{j+1}(Q_1, z_2+\delta_2+Q_2-\xi^{k_2})\right] p^1_{j+1,k_1} p^2_{j+1,k_2} \end{aligned} \tag{6.8.3}$$

$$H_j^{d'}(z_1+\delta_1, z_2+\delta_2) = \sum_{k_1:\xi^{k_1}> z_1+\delta_1} \sum_{k_2:\xi^{k_2}> z_2+\delta_2} \left[2c_{j+1,0} + f_{j+1}(z_1+\delta_1+Q_1-\xi^{k_1}, z_2+\delta_2+Q_2-\xi^{k_2})\right] p^1_{j+1,k_1} p^2_{j+1,k_2} \tag{6.8.4}$$

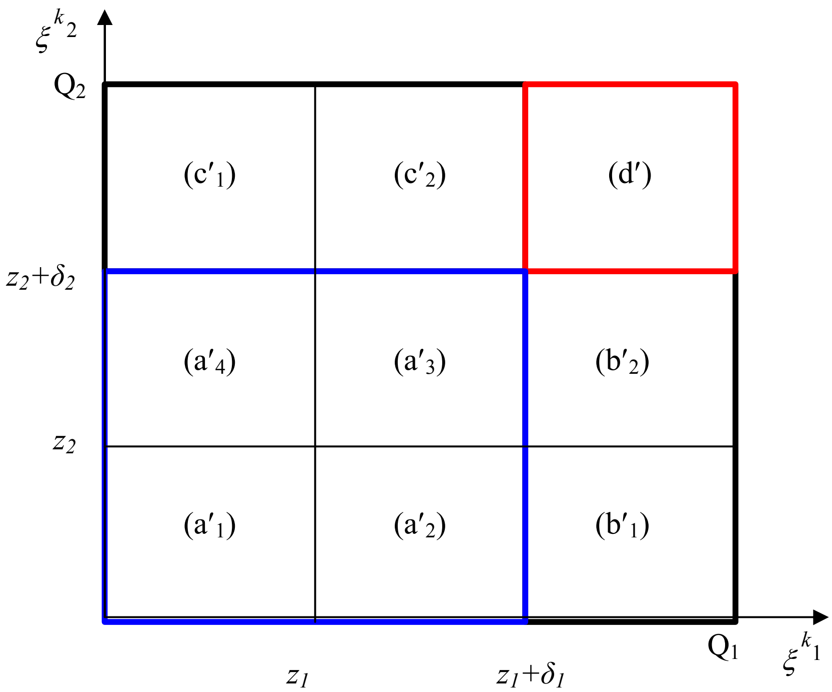


Figure 6.8. The definition space of $H_j(z_1+\delta_1, z_2+\delta_2)$.

Subtracting the terms that correspond to the same nine regions of Figures 6.7 and 6.8 we obtain:

$H_j(z_1, z_2) - H_j(z_1+\delta_1, z_2+\delta_2) =$

$$\left[H_j^{a}(z_1,z_2) - H_j^{a'_1}(z_1+\delta_1, z_2+\delta_2)\right] + \left[H_j^{b_1}(z_1,z_2) - H_j^{a'_2}(z_1+\delta_1, z_2+\delta_2)\right] + \left[H_j^{b_2}(z_1,z_2) - H_j^{b'_1}(z_1+\delta_1, z_2+\delta_2)\right] +$$

$$+\left[H_j^{c_1}(z_1,z_2) - H_j^{a'_4}(z_1+\delta_1, z_2+\delta_2)\right] + \left[H_j^{d_1}(z_1,z_2) - H_j^{a'_3}(z_1+\delta_1, z_2+\delta_2)\right] + \left[H_j^{d_2}(z_1,z_2) - H_j^{b'_2}(z_1+\delta_1, z_2+\delta_2)\right] +$$

$$\left[H_j^{c_2}(z_1,z_2) - H_j^{c'_1}(z_1+\delta_1, z_2+\delta_2)\right] + \left[H_j^{d_4}(z_1,z_2) - H_j^{c'_2}(z_1+\delta_1, z_2+\delta_2)\right] + \left[H_j^{d_3}(z_1,z_2) - H_j^{d'}(z_1+\delta_1, z_2+\delta_2)\right] =$$

$$= \left[c_{j,j+1} + \sum_{k_1:\xi^{k_1} \le z_1} \sum_{k_2:\xi^{k_2} \le z_2} f_{j+1}(z_1 - \xi^{k_1}, z_2 - \xi^{k_2}) p^1_{j+1,k_1} p^2_{j+1,k_2} \right.$$

$$\left. - c_{j,j+1} - \sum_{k_1:\xi^{k_1} \le z_1} \sum_{k_2:\xi^{k_2} \le z_2} f_{j+1}(z_1 + \delta_1 - \xi^{k_1}, z_2 + \delta_2 - \xi^{k_2}) p^1_{j+1,k_1} p^2_{j+1,k_2} \right]$$

$$+\Bigg[\sum_{k_1:z_1<\xi^{k1}\le z_1+\delta_1}\ \sum_{k_2:\xi^{k2}\le z_2}\Big[2c_{j+1,0}+f_{j+1}(z_1+Q_1-\xi^{k_1},Q_2)\Big]p^1_{j+1,k_1}p^2_{j+1,k_2}$$

$$-\sum_{k_1:z_1<\xi^{k1}\le z_1+\delta_1}\ \sum_{k_2:\xi^{k2}\le z_2}f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})p^1_{j+1,k_1}p^2_{j+1,k_2}\Bigg]$$

$$+\Bigg[\sum_{k_1:z_1+\delta_1<\xi^{k1}}\ \sum_{k_2:\xi^{k2}\le z_2}\Big[2c_{j+1,0}+f_{j+1}(z_1+Q_1-\xi^{k_1},Q_2)\Big]p^1_{j+1,k_1}p^2_{j+1,k_2}$$

$$-\sum_{k_1:\xi^{k1}>z_1+\delta_1}\ \sum_{k_2:\xi^{k2}\le z_2}\Big[2c_{j+1,0}+f_{j+1}(z_1+\delta_1+Q_1-\xi^{k_1},Q_2)\Big]p^1_{j+1,k_1}p^2_{j+1,k_2}\Bigg]$$

$$+\Bigg[\sum_{k_1:\xi^{k1}\le z_1}\ \sum_{k_2:z_2<\xi^{k2}\le z_2+\delta_2}\Big[2c_{j+1,0}+f_{j+1}(Q_1,z_2+Q_2-\xi^{k_2})\Big]p^1_{j+1,k_1}p^2_{j+1,k_2}$$

$$-\sum_{k_1:\xi^{k1}\le z_1}\ \sum_{k_2:z_2<\xi^{k2}\le z_2+\delta_2}f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})p^1_{j+1,k_1}p^2_{j+1,k_2}\Bigg]$$

$$+\Bigg[\sum_{k_1:z_1<\xi^{k1}\le z_1+\delta_1}\ \sum_{k_2:z_2<\xi^{k2}\le z_2+\delta_2}\Big[2c_{j+1,0}+f_{j+1}(z_1+Q_1-\xi^{k_1},z_2+Q_2-\xi^{k_2})\Big]p^1_{j+1,k_1}p^2_{j+1,k_2}$$

$$-\sum_{k_1:z_1<\xi^{k1}\le z_1+\delta_1}\ \sum_{k_2:z_2<\xi^{k2}\le z_2+\delta_2}f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})p^1_{j+1,k_1}p^2_{j+1,k_2}\Bigg]$$

$$+\Bigg[\sum_{k_1:z_1+\delta_1<\xi^{k1}}\ \sum_{k_2:z_2<\xi^{k2}\le z_2+\delta_2}\Big[2c_{j+1,0}+f_{j+1}(z_1+Q_1-\xi^{k_1},z_2+Q_2-\xi^{k_2})\Big]p^1_{j+1,k_1}p^2_{j+1,k_2}$$

$$-\sum_{k_1:\xi^{k1}>z_1+\delta_1}\ \sum_{k_2:z_2<\xi^{k2}\le z_2+\delta_2}\Big[2c_{j+1,0}+f_{j+1}(z_1+\delta_1+Q_1-\xi^{k_1},Q_2)\Big]p^1_{j+1,k_1}p^2_{j+1,k_2}\Bigg]$$

$$+\Bigg[\sum_{k_1:\xi^{k1}\le z_1}\ \sum_{k_2:z_2+\delta_2<\xi^{k2}}\Big[2c_{j+1,0}+f_{j+1}(Q_1,z_2+Q_2-\xi^{k_2})\Big]p^1_{j+1,k_1}p^2_{j+1,k_2}$$

$$-\sum_{k_1:\xi^{k1}\le z_1}\ \sum_{k_2:\xi^{k2}>z_2+\delta_2}\Big[2c_{j+1,0}+f_{j+1}(Q_1,z_2+\delta_2+Q_2-\xi^{k_2})\Big]p^1_{j+1,k_1}p^2_{j+1,k_2}\Bigg]$$

$$+\left[\sum_{k_1:z_1<\xi^{k1}\le z_1+\delta_1}\sum_{k_2:z_2+\delta_2<\xi^{k2}}\left[2c_{j+1,0}+f_{j+1}(z_1+Q_1-\xi^{k_1},z_2+Q_2-\xi^{k_2})\right]p^1_{j+1,k_1}p^2_{j+1,k_2}\right.$$

$$\left.-\sum_{k_1:z_1<\xi^{k1}\le z_1+\delta_1}\sum_{k_2:\xi^{k2}>z_2+\delta_2}\left[2c_{j+1,0}+f_{j+1}(Q_1,z_2+\delta_2+Q_2-\xi^{k_2})\right]p^1_{j+1,k_1}p^2_{j+1,k_2}\right]$$

$$+\left[\sum_{k_1:z_1+\delta_1<\xi^{k1}}\sum_{k_2:z_2+\delta_2<\xi^{k2}}\left[2c_{j+1,0}+f_{j+1}(z_1+Q_1-\xi^{k_1},z_2+Q_2-\xi^{k_2})\right]p^1_{j+1,k_1}p^2_{j+1,k_2}\right.$$

$$\left.-\sum_{k_1:\xi^{k1}>z_1+\delta_1}\sum_{k_2:\xi^{k2}>z_2+\delta_2}\left[2c_{j+1,0}+f_{j+1}(z_1+\delta_1+Q_1-\xi^{k_1},z_2+\delta_2+Q_2-\xi^{k_2})\right]p^1_{j+1,k_1}p^2_{j+1,k_2}\right] \qquad (6.9)$$

In Eq. (6.9) we can distinguish two types of terms. The first type includes the first, third, sixth, seventh, eighth, and ninth terms. Considering the first term we have:

$$H_j^a(\mathrm{z}_1,\mathrm{z}_2)-H_j^{a'_1}(\mathrm{z}_1+\delta_1,\mathrm{z}_2+\delta_2) =$$

$$=\sum_{k_1:\xi^{k1}\le z_1}\sum_{k_2:\xi^{k2}\le z_2}[f_{j+1}(z_1-\xi^{k_1},z_2-\xi^{k_2})-f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})]p^1_{j+1,k_1}p^2_{j+1,k_2}$$

Since $f_{j+1}(z_1, z_2)$ is monotonically non-increasing, this term is non-negative. Similarly we can show that the differences corresponding to the third, sixth, seventh, eighth, and ninth terms are also non-negative.

The second type of terms includes the second, forth and fifth terms. Considering the second term we have:

$$H_j^{b_1}(\mathrm{z}_1,\mathrm{z}_2)-H_j^{a'_2}(\mathrm{z}_1+\delta_1,\mathrm{z}_2+\delta_2)=$$

$$=\sum_{k_1:z_1<\xi^{k1}\le z_1+\delta_1}\sum_{k_2:\xi^{k2}\le z_2}\left[2c_{j+1,0}+f_{j+1}(z_1+Q_1-\xi^{k_1},Q_2)-f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})\right]p^1_{j+1,k_1}p^2_{j+1,k_2}$$

Recall that from **LEMMA 1:**

$$f_j(z_1, z_2) \le f_j(Q_1, Q_2)+ 2c_{0j} \quad \textit{for all } z \in S_j$$

therefore:

$$f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})\le f_{j+1}(Q_1,Q_2)+2c_{j+1,0}$$

However,

$$f_{j+1}(Q_1, Q_2) \leq f_{j+1}(z_1 + Q_1 - \xi^{k_1}, Q_2)$$

since $\xi^{k_1} > z_1$ and f_{j+1} is non-increasing. Thus:

$$f_{j+1}(z_1 + \delta_1 - \xi^{k_1}, z_2 + \delta_2 - \xi^{k_2}) \leq f_{j+1}(Q_1, Q_2) + 2c_{j+1,0} \leq f_{j+1}(z_1 + Q_1 - \xi^{k_1}, Q_2) + 2c_{j+1,0}$$

Therefore $H_j^{b_1}(z_1, z_2) - H_j^{a'_2}(z_1 + \delta_1, z_2 + \delta_2) \geq 0$. Similarly we can show that the forth and fifth terms of Eq. (6.9) are also non-negative.

Considering the non-negativity of all terms of Eq. (6.9) it is clear that $H_j(z_1, z_2) - H_j(z_1+\delta_1, z_2+\delta_2) \geq 0$ and hence, $H_j(z_1, z_2)$ is a non-increasing function.

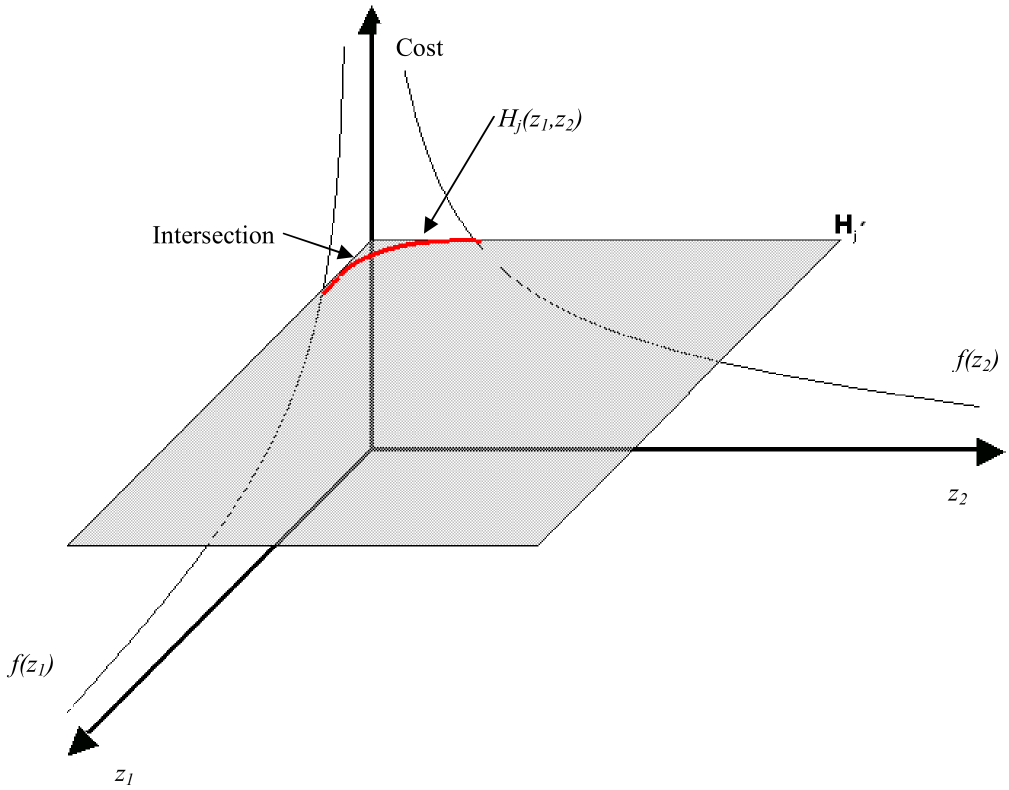


Figure 6.9. The combined threshold graphical representation.

Note now that part (b) $H'_j(z_1,z_2)$ of Eq. (6.2) is independent of z_1, z_2 and, thus, a constant $H'_j(z_1,z_2) = H'_j$ in the z_1, z_2 space. Figure 6.9 plots the terms $H_j(z_1,z_2)$ and H'_j with $H_j(z_1,z_2)$ been non-increasing as shown above. From this Figure it is clear that $f_j(z_1,z_2) = \min\{H_j(z_1,z_2), H'_j(z_1,z_2)\}$ is a non-increasing function.

Furthermore, the intersection $H_j(z_1,z_2) \cap H'_j$ can in general be described by a function of the form $h_j(z_1,z_2) = c$. This is more clearly shown in Figure 6.10.

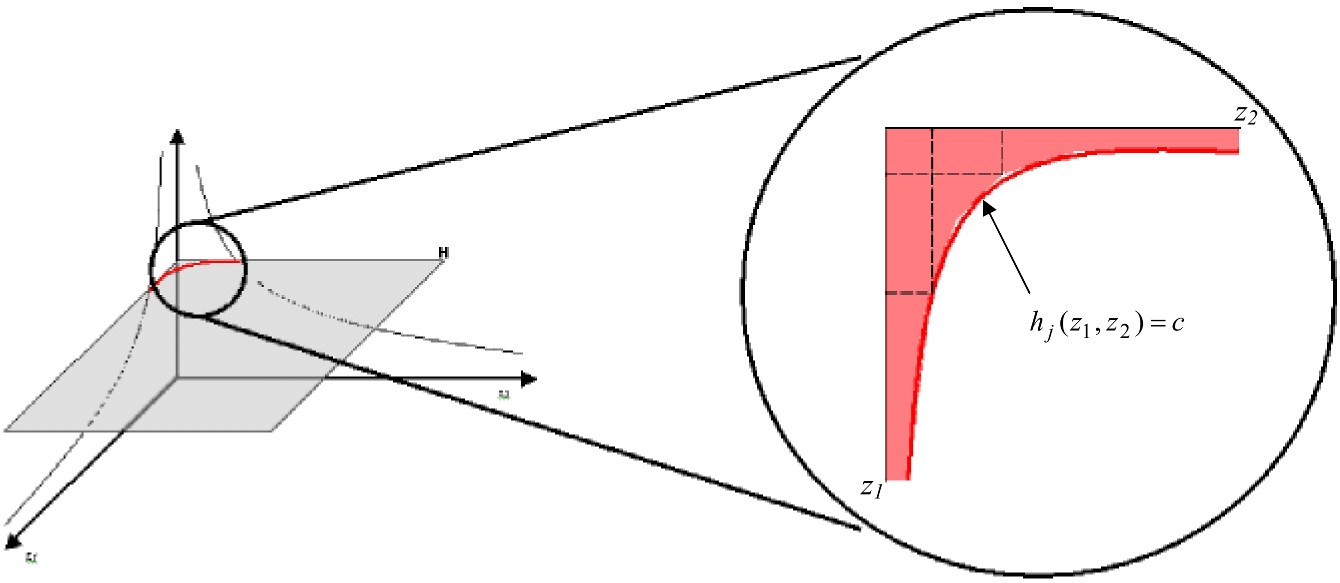


Figure 6.10. A better view of the threshold function.

Every (z_1,z_2) combination that lies within the highlighted area, $h_j(z_1,z_2) < c_j$, corresponds to a depot return before visiting customer $j+1$. For (z_1,z_2) combinations that lie outside the highlighted area $h_j(z_1,z_2) \geq c_j$, the vehicle should proceed to the next customer. This concludes the proof of Theorem 1.

This theorem provides the optimal routing policy; i.e. the policy that if followed by the vehicle, we will obtain the minimum expected value of the travel cost.

6.2.3 Solution Algorithm

In order to solve the compartmentalized case of multiple product delivery, we developed an appropriate algorithm that uses Dynamic Programming to derive the optimal solution in a reasonable amount of time. Based on the formulation presented in Section 6.2.1, the algorithm starts from the end of the route (last customer to be visited) and iterates towards the beginning of the route, calculating the remaining minimum expected cost from each customer site until the end of the route. This procedure computes the minimum expected cost of the route, given a distance matrix, and demand probability mass functions. Based on the result of the algorithm, the threshold function $h_j(z_1, z_2) = c$ for each customer j can be obtained, in line with what has already been described in Section 6.2.2 to provide the optimal routing policy. A characteristic example follows below.

Consider the 5-customer network of Figure 6.11. The vehicle capacity is $Q = 10$ units and is equally split between two products ($Q_1 = Q_2 = 5$); the demand ξ_j^i for each product i (i = 1,2) and customer j (j = 1, …, 5) is given in Appendix C, and the distances between the nodes c_{ij} are given in Figure 6.11.

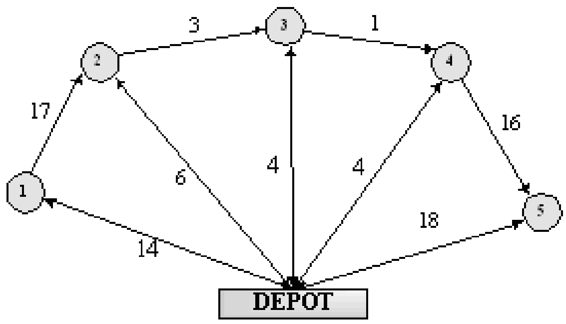


Figure 6.11. 5-customer network for the multiple-product extension.

The problem is solved using the dynamic programming algorithm. Let (z_1, z_2) be the remaining quantities of each product type in the vehicle. Let $V_j(z_1, z_2)$ and $x_j(z_1, z_2)$ be the minimum expected cost and the corresponding decision after customer j has been served. Clearly, $V_5(z_1, z_2) = 18, x_5(z_1, z_2) = 1$ for $z_1 \in \{0, \ldots, 5\}, z_2 \{0,...,5\}$. In Tables 6.1-6.5 we provide the results for nodes 4, 3, 2, 1, 0. In these Tables,

(z_1, z_2) represents the quantity carried by the vehicle after customer j has been served; each cell includes two values: The first is the value of $x_j(z_1, z_2)$ and the second is the value of $V_j(z_1, z_2)$.

Table 6.1. Results obtained for node 4.

z2 \ z1	0	1	2	3	4	5
0	1; 40	1; 40	1; 40	1; 40	1; 40	1; 40
1	1; 40	1; 40	1; 40	1; 40	1; 40	1; 40
2	1; 40	1; 40	1; 40	1; 40	1; 40	1; 40
3	1; 40	1; 40	1; 40	1; 40	1; 40	1; 40
4	1; 40	1; 40	1; 40	0; 37,6	0; 34,7	0; 34,4
5	1; 40	1; 40	1; 40	0; 37,2	0; 34,4	0; 34,0

Table 6.2. Results obtained for node 3.

z2 \ z1	0	1	2	3	4	5
0	1; 45,4	0; 45,4	0; 44,7	0; 44,5	0; 44,4	0; 44,4
1	1; 45,4	0; 43,9	0; 42,8	0; 42,1	0; 41,9	0; 41,9
2	1; 45,4	0; 43,5	0; 42,3	0; 41,6	0; 41,5	0; 41,4
3	1; 45,4	0; 43,3	0; 42,1	0; 41,4	0; 41,2	0; 41,1
4	1; 45,4	0; 43,2	0; 41,9	0; 41,3	0; 41,0	0; 40,9
5	1; 45,4	0; 43,2	0; 41,9	0; 41,2	0; 39,9	0; 38,4

Table 6.3. Results obtained for node 2.

z2 \ z1	0	1	2	3	4	5
0	1; 52,0	1; 52,0	1; 52,0	1; 52,0	1; 52,0	1; 52,0
1	1; 52,0	1; 52,0	1; 52,0	1; 52,0	1; 52,0	1; 52,0
2	1; 52,0	0; 51,9	0; 51,8	0; 51,7	0; 51,6	0; 51,6
3	1; 52,0	0; 51,3	0; 50,9	0; 50,7	0; 50,5	0; 50,4
4	0; 51,47	0; 49,3	0; 48,4	0; 47,5	0; 47,0	0; 46,8
5	0; 51,46	0; 49,3	0; 47,5	0; 46,2	0; 45,4	0; 44,9

Table 6.4. Results obtained for node 1.

z1 / z2	0	1	2	3	4	5
0	1; 70,2	1; 70,2	1; 70,2	1; 70,2	1; 70,2	1; 70,2
1	1; 70,2	1; 70,2	1; 70,2	1; 70,2	1; 70,2	1; 70,2
2	1; 70,2	1; 70,2	1; 70,2	1; 70,2	1; 70,2	1; 70,2
3	1; 70,2	1; 70,2	1; 70,2	0; 70,0	0; 69,6	0; 69,5
4	1; 70,2	1; 70,2	0; 70,0	0; 69,3	0; 68,7	0; 68,6
5	1; 70,2	1; 70,2	0; 69,5	0; 68,4	0; 67,6	0; 67,2

$H_1 > H'_1$

$H_1 \leq H'_1$

Table 6.5. Result obtained for node 0.

z1 / z2	5
5	84,0

Therefore the minimum expected total cost is equal to 84,0.

Figure 6.12 illustrates the threshold function $h_1(z_1, z_2) = c$ as obtained from the results of Table 6.4 (for Customer 1). It is clearly seen from Table 6.4 that H'_1 = 70,2. For all (z_1,z_2) pairs with $H_1(z_1,z_2) \leq H'_1$ the vehicle should proceed to the next customer directly.

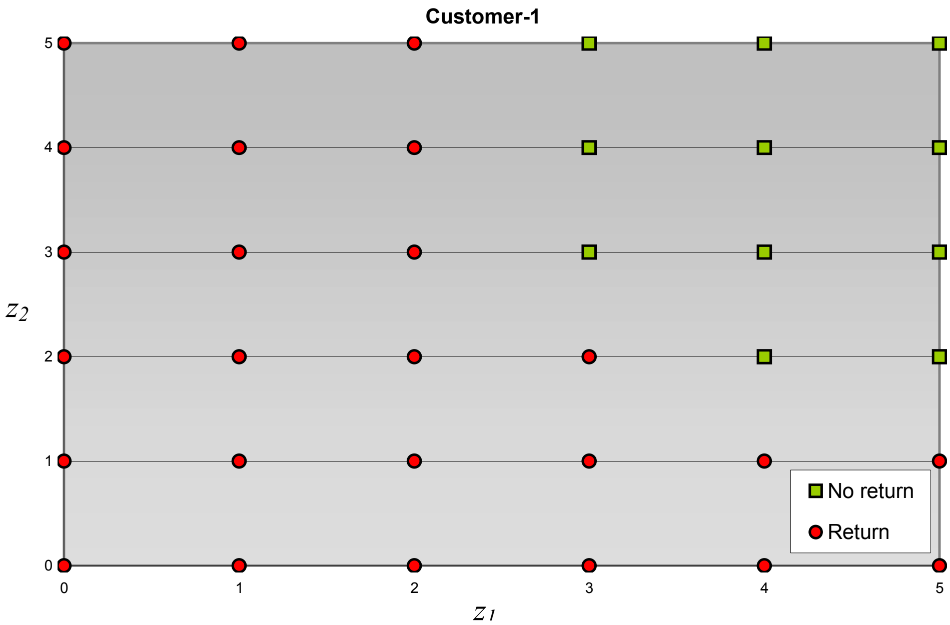


Figure 6.12. The corresponding load combinations after serving customer 1.

Figure 6.13 illustrates the threshold function $h_6(z_1,z_2)$ of customer-6 in a different example with $Q_1=Q_2=10$.

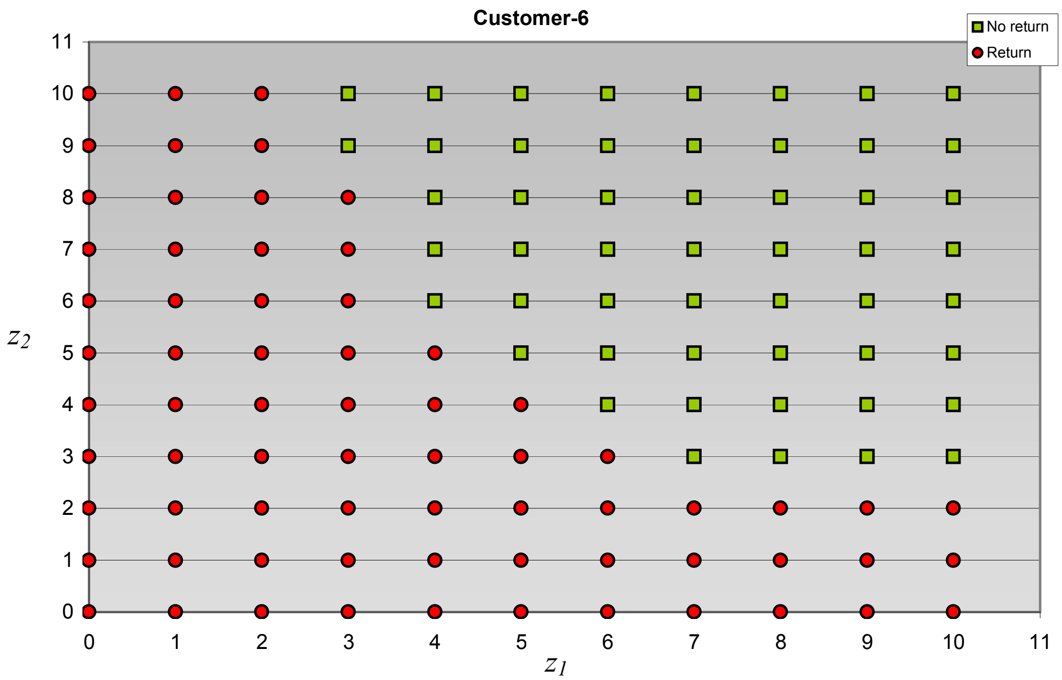


Figure 6.13. The corresponding load combinations after serving customer 6.

This figure clearly demonstrates the existence of $h_6(z_1,z_2)$ according to which for any combination below the border defined by the switch between the red and green points, the optimal decision is to return to the depot in order to refill the vehicle. Conversely, for any combinations above this border the optimal decision would be to proceed directly to the next customer. This policy can be very simply and clearly communicated to the vehicle driver and result in significant cost savings for the fleet operator. The performance of the algorithm was found to be within acceptable levels. As an indication, for a test problem of 10 customer points, 2 products and vehicle capacity 10 units per product, the algorithm derived the minimum expected cost, as well as the threshold curves per customer, within 9 sec. The number of combinations examined for each of the customer points 2-9 were approximately 1330. The experiments were run on a PC equipped with Intel Pentium IV, at 2.4 GHz and 512 MB of RAM.

In order to further assess the performance of the algorithm for the two-product case a large number of problem test cases were created and run. We generated approximately 30,000 problems of appropriate characteristics and the results obtained are shown in Figure 6.14.

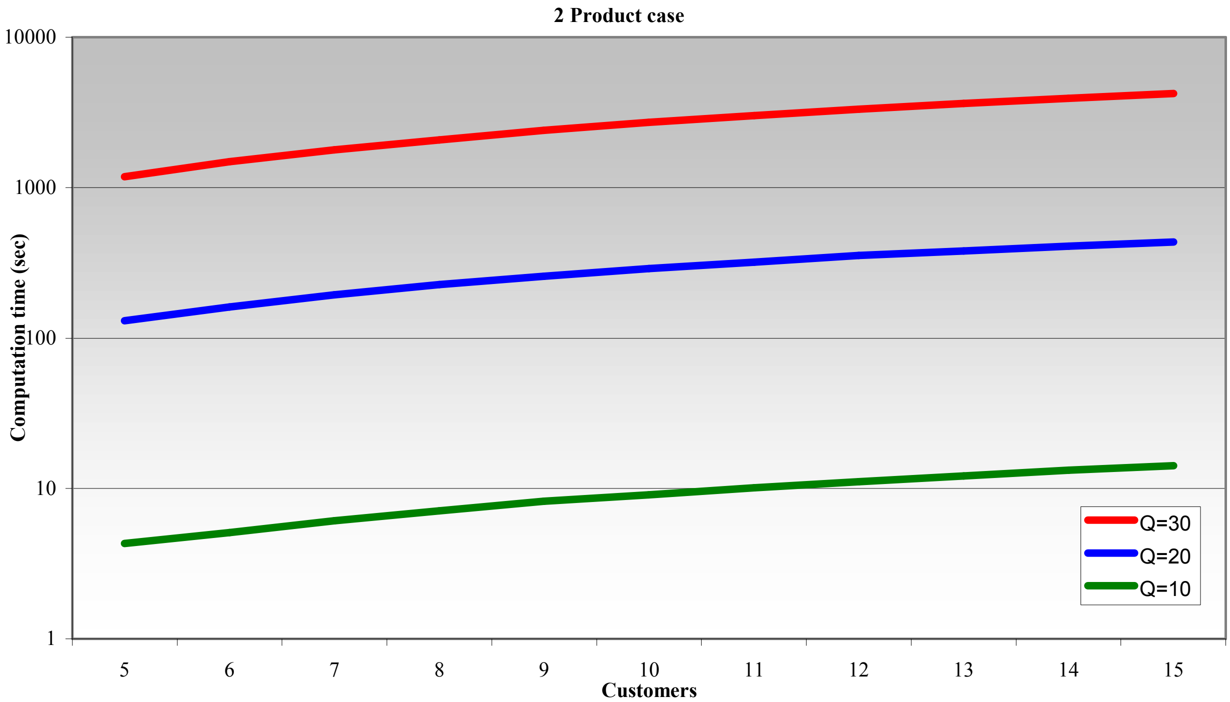


Figure 6.14. Performance results of the algorithm.

Three different problem test cases are presented in this figure. The first test case (10,000 randomly generated problems) concerns a vehicle with total capacity Q = 10 equally split between the two compartments, and is shown in green color. The second test case (another 10,000 randomly generated problems) concerns a vehicle with total capacity Q = 20 equally split between the two compartments, and is shown in blue color. The third test case (another 10,000 randomly generated problems) concerns a vehicle with total capacity Q = 30 equally split between the two compartments, and is shown in red color. Each point in these curves corresponds to the average solution time for 1,000 randomly generated problems. The demand distributions were generated randomly.

From Figure 6.14 it can be clearly seen that for a given number of customers, the increase of the vehicle capacity (Q = 10, 20, 30) results in a significant increase in the computational time of the algorithm (note that the scale of the y-axis is logarithmic). On the other hand, if the capacity of the vehicle is kept constant, and the number of customers is increased, the computational time also increases, almost linearly. In order to analyze this latter relationship further, ten different test cases were run with Q = 20, ranging from five to fifty customer points (1000 randomly generated problems for each customer) and are shown in Figure 6.15. Each point in this Figure represents the average time for the 1000 problems of the particular problem set. The increase in the computational time is indeed linear with the number of customers. Even at the 50-customer instance the algorithm took approximately 1635 seconds (= 27 minutes) in order to obtain the solution of the problem.

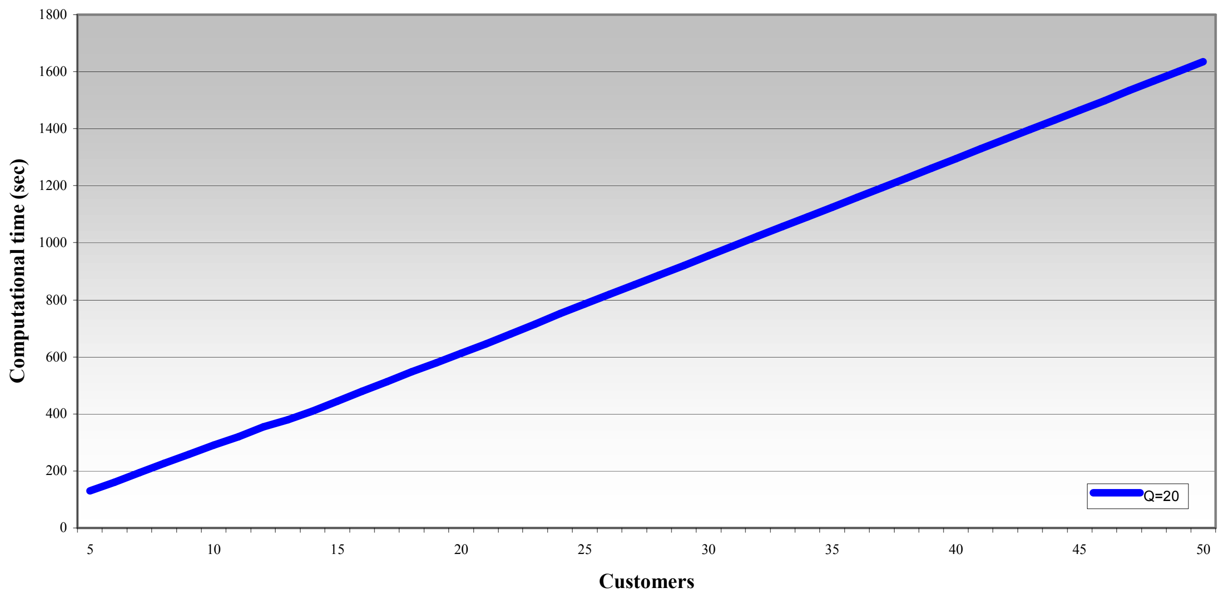


Figure 6.15. Performance results with up to 50 customer points.

6.3 Multiple Product Delivery: Unified Load

Recall from Chapter 4 that in the unified load case (shown in Figure 4.1) the vehicle may carry any quantity of product $i \in \{1, \dots, K\}$, provided that the total capacity Q of the vehicle is not exceeded.

Upon completion of service at customer site j, the vehicle's driver has to make the same decision as the one described in Sections 5.2 and 6.2, i.e. proceed to customer $j+1$, or return to the depot in order to refill the vehicle and resume the route to serve customer $j+1$. The objective of the problem is to serve all customers (replenish their stock for all items) and minimize travel cost.

As before, we declare ξ_j^i the stochastic demand of customer $j \in \{1, \dots, n\}$ for product i. The demand per customer is no longer independent with respect to the product types, since for each customer j $\xi_j^{k_1}+\dots+\xi_j^{k_k} \leq Q$. Thus, the probability mass function $P_j(z_1, z_2, \dots, z_K)$ is, in general, joint.

6.3.1 Dynamic Programming Formulation

We will focus on the 2 product case, which can be extended to $k>2$ products in a straightforward manner. Let z_i represent the stock onboard for product i after serving customer j. Let $p_j(k_1, k_2) = \text{Prob}(\xi_j^1 = \xi^{k_1}, \xi_j^2 = \xi^{k_2})$ represent the combined probability mass function. Also let $f_j(z_1, z_2)$ denote the minimum expected cost from customer j onward. Note that all product quantities are calculated using the same unit of measure e.g. m^3 or kg.

In this case there are additional issues to be considered. First, upon visiting the depot, an additional decision needs to be made regarding the quantities of stock to be loaded onto the vehicle. Let θ be the quantity of product type-1 to be loaded onto the vehicle. Then in the *2*-product case, the quantity of product type-2 will be $Q-\theta$. Secondly, two sequential returns to the depot may occur in this problem. This may occur only if the vehicle does not proceed to client $j+1$ but visits the depot first. In this case if the quantity θ for product type-1 (or $Q-\theta$ for product type-2) loaded onto the vehicle at the depot is not adequate to fully satisfy the client's demand, the vehicle will return to the depot once more, and make an additional, informed this time, decision.

The stock s loaded to the vehicle for product type-1 (and Q-s for product type-2) will guarantee that the demand of customer $j+1$ is fully satisfied. This is shown in Figure 6.16 for the case where $\theta < \xi^{k_1}$.

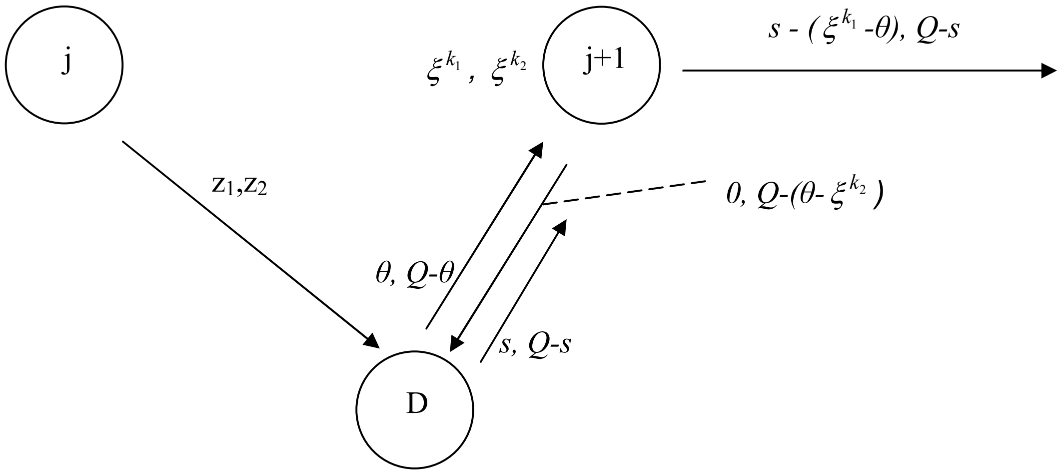


Figure 6.16. The case that $\theta < \xi_1^{k_1}$.

The mathematical formulation of the unified load problem (the proof of which is given in Appendix F) can be obtained using part (b) of Figure (6.1) and (6.16) and is shown below:

$$f_j(z_1, z_2) = \min$$

$$\left\{\begin{array}{l}
\left.\begin{array}{ll}
c_{j,j+1} + \sum\limits_{k_1:\xi^{k_1}\le z_1} \sum\limits_{k_2:\xi^{k_2}\le z_2} f_{j+1}(z_1-\xi^{k_1}, z_2-\xi^{k_2}) p_{j+1}(k_1,k_2) + & \text{(term a)} \\
+ \sum\limits_{k_1:z_1<\xi^{k_1}} \sum\limits_{k_2:\xi^{k_2}\le z_2} \left[2c_{j+1,0} + \min\limits_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}(\theta-(\xi^{k_1}-z_1), Q-\theta)\right] p_{j+1}(k_1,k_2) + & \text{(term b)} \\
+ \sum\limits_{k_1:\xi^{k_1}\le z_1} \sum\limits_{k_2:z_2<\xi^{k_2}} \left[2c_{j+1,0} + \min\limits_{\xi^{k_2}-z_2\le\theta\le Q} f_{j+1}(\theta, Q-\theta-(\xi^{k_2}-z_2))\right] p_{j+1}(k_1,k_2) + & \text{(term c)} \\
+ \sum\limits_{k_1:z_1<\xi^{k_1}} \sum\limits_{k_2:z_2<\xi^{k_2}} \left[2c_{j+1,0} + \min\limits_{\xi^{k_1}-z_1\le\theta\le Q-(\xi^{k_2}-z_2)} f_{j+1}(\theta-(\xi^{k_1}-z_1), Q-\theta-(\xi^{k_2}-z_2))\right] p_{j+1}(k_1,k_2) & \text{(term d)}
\end{array}\right\} \text{(part a)} \\
\\
\left.\min\limits_{0\le\theta\le Q}\left\{\begin{array}{c}
c_{j,0} + c_{0,j+1} + \\
\sum\limits_{\xi^{k_1}\le\theta} \sum\limits_{\xi^{k_2}\le Q-\theta} f_{j+1}(\theta-\xi^{k_1}, Q-\theta-\xi^{k_2}) p_{j+1}(k_1,k_2) + \\
+ \sum\limits_{\theta<\xi^{k_1}} \sum\limits_{\xi^{k_2}\le Q-\theta} \left[2c_{j+1,0} + \min\limits_{\xi^{k_1}-\theta\le s\le Q} f_{j+1}(s-(\xi^{k_1}-\theta), Q-s)\right] p_{j+1}(k_1,k_2) + \\
+ \sum\limits_{\xi^{k_1}\le\theta} \sum\limits_{Q-\theta<\xi^{k_2}} \left[2c_{j+1,0} + \min\limits_{0\le s\le 2Q-\theta-\xi^{k_2}} f_{j+1}\{s, Q-s-[\xi^{k_2}-(Q-\theta)]\}\right] p_{j+1}(k_1,k_2)
\end{array}\right\}\right\} \text{(part b)}
\end{array}\right. \qquad (6.10)$$

In Eq. (6.10) the first part in the minimization represents the expected cost of going directly to the next customer, whereas the second part represents the expected cost of the restocking action. As it can be clearly observed, the first part of the minimization equation includes four terms, which correspond to the four cases in part (b) of Figure 6.1.

The first term (after $c_{j,j+1}$) represents the expected cost incurred if the vehicle proceeds to the next customer ($j+1$) directly and the stock of both products is sufficient to fully satisfy the demands ξ^{k_1}, ξ^{k_2} of the next customer (Area A in part (b) of Figure 6.1). The loads of the vehicle before and after serving customer $j+1$ are schematically represented in Figure 6.17 below.

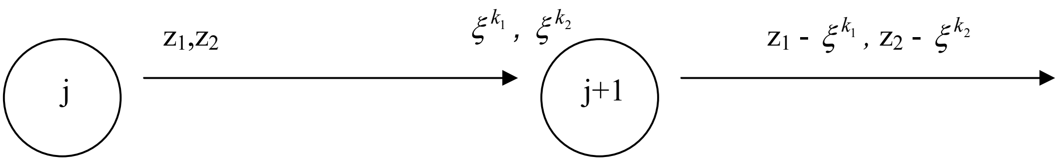


Figure 6.17. The case in which the demand is fully satisfied: $z_1 \geq \xi^{k_1}$, $z_2 \geq \xi^{k_2}$.

The second term represents the expected cost incurred if the vehicle proceeds to the next customer directly and the stock on board of product type-1 is not sufficient to fully satisfy the demand of the next customer, while the stock on board of product type-2 is sufficient (Area B in part (b) of Figure 6.1). The path of the vehicle and the corresponding vehicle loads are shown in Figure 6.18 below.

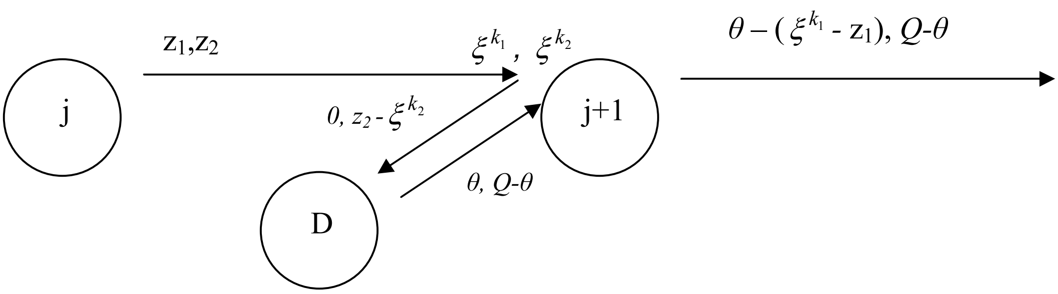


Figure 6.18. The case in which $z_1 < \xi^{k_1}$ and $z_2 \geq \xi^{k_2}$.

In this case, the vehicle will visit customer $j+1$, it will fully satisfy the demand ξ^{k_2} for product type-2 but will partially satisfy the demand ξ^{k_1} for product type-1, and will, therefore, return to the depot for stock replenishment. There, it will load θ units of measure for product type-1 and Q-θ for product type-2. The stock left on board after serving customer $j+1$ is $\theta - (\xi^{k_1} - z_1)$ for product type-1 and Q-θ for product type-2. Note that always $\theta \geq \xi^{k_1} - z_1$, since the exact value of ξ^{k_1} is known upon visiting the customer. Note also the additional minimization inside the parenthesis the purpose of which is to select the optimal θ (lower cost incurred) for this stage of the algorithm.

The third term represents the cost incurred if the vehicle proceeds to the next customer directly and the stock on board of product type-1 is sufficient to fully satisfy the demand ξ^{k_1} of the next customer, while the stock of product type-2 is not sufficient to satisfy ξ^{k_2} (Area C in part (b) of Figure 6.1). This case is analogous to the previous one; the path of the vehicle and the corresponding vehicle loads are shown in Figure 6.19. Note also the additional minimization inside the parenthesis whose purpose is to select the optimal θ (lower cost incurred) for that stage of the algorithm.

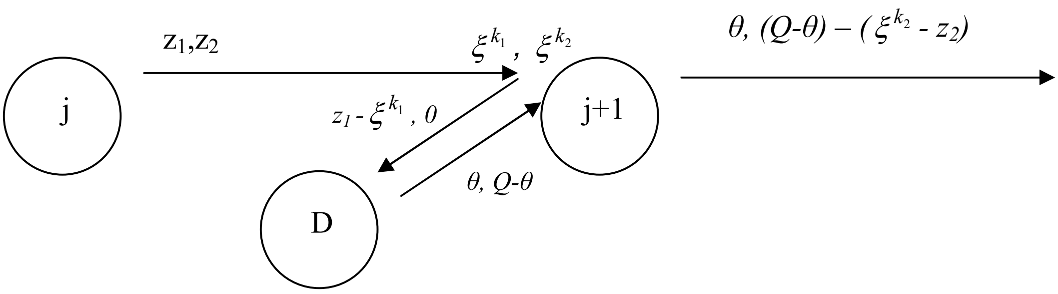


Figure 6.19. The case in which $z_1 \geq \xi^{k_1}$ and $z_2 < \xi^{k_2}$.

The fourth term represents the cost incurred if the vehicle goes to the next customer directly and neither the stock of product type-1 nor the stock of product type-2 are sufficient to fully satisfy the demand ξ^{k_1}, ξ^{k_2} (Area D in part (b) of Figure 6.1). The path of the vehicle and the corresponding vehicle loads are shown in Figure 6.20. Note also the additional minimization inside the parenthesis whose purpose is to select the optimal θ (lower cost incurred) for that stage of the algorithm.

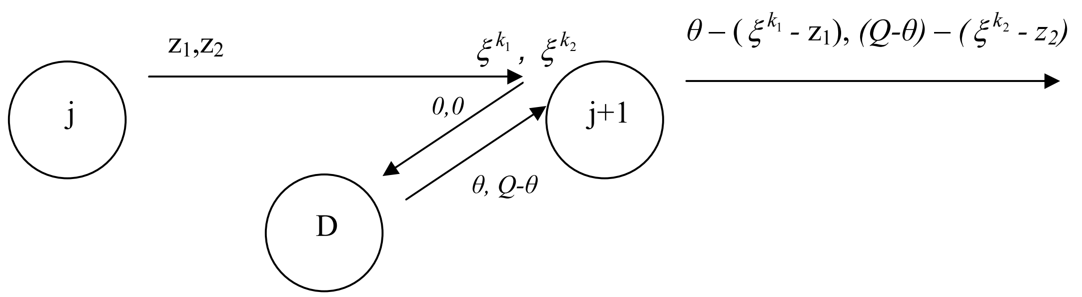


Figure 6.20. The case in both $z_1 < \xi^{k_1}$ and $z_2 < \xi^{k_2}$.

The second part of the minimization equation Eq. (6.10) consists of three terms (after the sum $c_{j,0} + c_{0,j+1}$). Note the additional external minimization of the entire part (b) of Eq. (6.10) whose purpose is to select the optimal θ (lower cost incurred) for that stage of the algorithm. Due to the fact that this is a proactive depot return, the values of z_1 and z_2 do not affect the result, which is thus independent of z_1 and z_2. Furthermore, since ξ^{k_1} and ξ^{k_2} were not known the quantities θ and $Q-\theta$ loaded at the depot may not be sufficient to satisfy the demand of customer $j+1$. The first term of the second part represents the case in which the vehicle visits the depot, loads θ of product type-1 and $Q-\theta$ of product type-2, proceeds to customer $j+1$ and can fully satisfy demand ξ^{k_1}, ξ^{k_2}. This is shown in Figure 6.21.

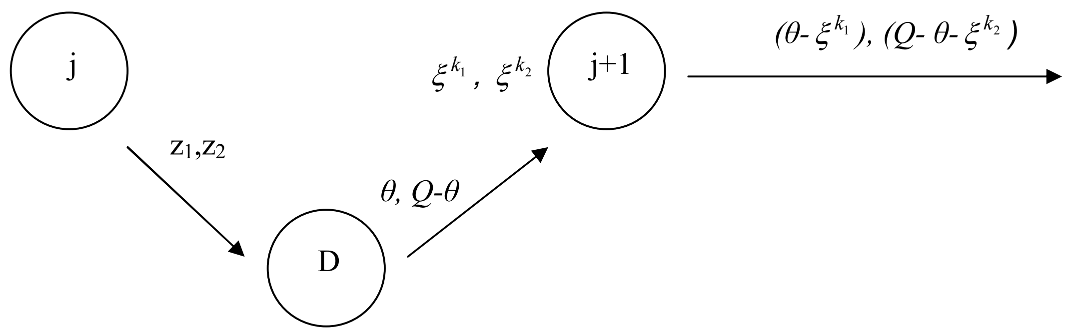


Figure 6.21. The case in which $\theta \geq \xi_1^{k_1}$ and Q- $\theta \geq \xi_2^{k_2}$.

The second term represents the case in which upon visiting customer $j+1$ the vehicle cannot satisfy the customer's demand for product type-1. In this case, the vehicle will return to the depot once more, and make an additional, informed this time, decision for loading stock s for product type-1 (and $Q-s$ for product type-2) in order to guarantee that it will fully satisfy the demand of customer $j+1$ (see Figure 6.16). In addition to the external minimization with respect to θ, note the internal minimization (inside the parenthesis) with respect to s, whose purpose is to select the optimal s (lower cost incurred) for that stage of the algorithm.

The third term represents the case that after visiting customer $j+1$ the vehicle cannot satisfy the customer's demand for product type-2. The course of action here is analogous to the one discussed previously. This is shown in Figure 6.22. In addition to the external minimization with respect to θ, note the internal minimization (inside the parenthesis) with respect to s, whose purpose is to select the optimal s (lower cost incurred) for that stage of the algorithm.

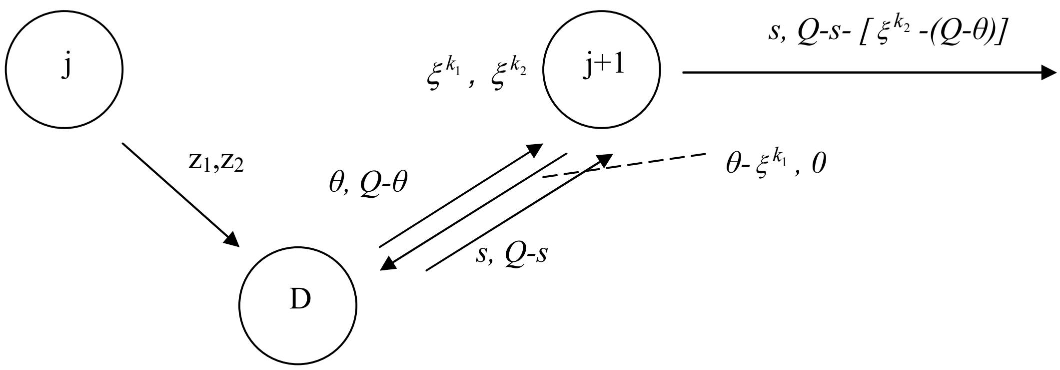


Figure 6.22. The case that $Q\text{-}\theta < \xi^{k_2}$.

6.3.2 Problem Characteristics

The value of the minimum expected cost can be estimated using Eq. (6.10). However, in order to develop the policy which leads to achieving this minimum expected cost for the unified load case we need to investigate a threshold theorem analogous to Theorem 1 of the compartmentalized load case. Proceeding in the same fashion as in the latter case we first introduce Lemma 2 and then Theorem 2.

LEMMA 2.

$$f_j(z_1,z_2) \le 2c_{0j} + \min_{0\le\theta\le Q} f_j(\theta, Q-\theta) \qquad \textit{for all } z_1,z_2 \in S_j \tag{6.11}$$

Proof. From Eq. (6.10) we obtain:

$$f_j(z_1,z_2) \le \min_{0\le\theta\le Q}\left\{\begin{array}{l} c_{j,0}+c_{0,j+1}+ \\ \displaystyle\sum_{k_1:\xi^{k_1}\le\theta}\ \sum_{k_2:\xi^{k_2}\le Q-\theta} f_{j+1}(\theta-\xi^{k_1},Q-\theta-\xi^{k_2})p_{j+1}(k_1,k_2)+ \\ \displaystyle +\sum_{k_1:\theta<\xi^{k_1}}\ \sum_{k_2:\xi^{k_2}\le Q-\theta}\left[2c_{j+1,0}+\min_{\xi^{k_1}-\theta\le s\le Q} f_{j+1}(s-(\xi^{k_1}-\theta),Q-s)\right]p_{j+1}(k_1,k_2)+ \\ \displaystyle +\sum_{k_1:\xi^{k_1}\le\theta}\ \sum_{k_2:Q-\theta<\xi^{k_2}}\left[2c_{j+1,0}+\min_{\xi^{k_2}-(Q-\theta)\le s\le Q} f_{j+1}(s,Q-s-(\xi^{k_2}-(Q-\theta)))\right]p_{j+1}(k_1,k_2) \end{array}\right\} \quad (6.12)$$

For $z_1 = \theta$ and $z_2 = Q\text{-}\theta$ part (a) of Eq. (6.10) becomes:

$$f_j^{(a)}(\theta,Q-\theta) = \left\{\begin{array}{l} \displaystyle c_{j,j+1}+\sum_{k_1:\xi^{k_1}\le\theta}\ \sum_{k_2:\xi^{k_2}\le Q-\theta} f_{j+1}(\theta-\xi^{k_1},Q-\theta-\xi^{k_2})p_{j+1}(k_1,k_2)+ \\ \displaystyle +\sum_{k_1:\theta<\xi^{k_1}}\ \sum_{k_2:\xi^{k_2}\le Q-\theta}\left[2c_{j+1,0}+\min_{\xi^{k_1}-\theta\le s\le Q} f_{j+1}(s-(\xi^{k_1}-\theta),Q-s)\right]p_{j+1}(k_1,k_2)+ \\ \displaystyle +\sum_{k_1:\xi^{k_1}\le\theta}\ \sum_{k_2:Q-\theta<\xi^{k_2}}\left[2c_{j+1,0}+\min_{\xi^{k_2}-(Q-\theta)\le s\le Q} f_{j+1}(s,Q-s-(\xi^{k_2}-(Q-\theta)))\right]p_{j+1}(k_1,k_2)+ \end{array}\right\} \quad (6.13)$$

Regarding the above function, note that in [terms (b) and (c) of Eq. (6.10)] the quantity of product-1 loaded to the vehicle upon its return to the depot is denoted by s (not to be confused with $z_1 = \theta$). Furthermore term (d) of Eq. (6.10) is zero since $P_{j+1}\left(\xi^{k_1}+\xi^{k_2} > \theta+Q-\theta = Q\right) = 0$.

From Eq. (6.10) and for $z_1 = \theta$ and $z_2 = Q\text{-}\theta$ the minimum value of $f_j(\theta,Q-\theta)$ with respect to $0 \le \theta \le Q$, i.e. $\min_{0\le\theta\le Q} f_j(\theta,Q-\theta)$ is given by part (a).

$$\min_{0\le\theta\le Q} f_j(\theta,Q-\theta) = \min_{0\le\theta\le Q} f_j^{(a)}(\theta,Q-\theta) \quad (6.14)$$

This is true since in this case $\min_{0\le\theta\le Q} f_j^{(a)}(\theta,Q-\theta) \le \min_{0\le\theta\le Q} f_j^{(b)}(\theta,Q-\theta)$.

The left-hand side of the above inequality is given by considering the minimum value of Eq. (6.13) with respect to $0 \le \theta \le Q$, while the right-hand side is part (b) of Eq. (6.10). This inequality holds since $c_{j,j+1} \le c_{j,0} + c_{0,j+1}$.

From Eqs. (6.14) and (6.13) we obtain:

$$\min_{0\le\theta\le Q} f_j(\theta, Q-\theta) =$$

$$= \min_{0\le\theta\le Q} \left\{ \begin{array}{l} c_{j,j+1} + \sum\limits_{\xi^{k_1}\le\theta} \sum\limits_{\xi^{k_2}\le Q-\theta} f_{j+1}(\theta-\xi^{k_1}, Q-\theta-\xi^{k_2}) p_{j+1}(k_1,k_2) + \\ + \sum\limits_{\xi^{k_1}>\theta} \sum\limits_{\xi^{k_2}\le Q-\theta} \left[2c_{j+1,0} + \min\limits_{\xi^{k_1}-\theta\le\theta\le Q} f_{j+1}(\theta-(\xi^{k_1}-\theta), Q-\theta) \right] p_{j+1}(k_1,k_2) + \\ + \sum\limits_{\xi^{k_1}\le\theta} \sum\limits_{\xi^{k_2}>Q-\theta} \left[2c_{j+1,0} + \min\limits_{\xi^{k_2}-(Q-\theta)\le\theta\le Q} f_{j+1}(Q-\theta, \theta-(\xi^{k_2}-(Q-\theta))) \right] p_{j+1}(k_1,k_2) + \end{array} \right\} \quad (6.15)$$

Note that the sum of the last three terms of the right-hand side of Eq. (6.15) is identical to the sum of the last three terms of the right-hand side of Eq. (6.12). By substituting the former to the latter we obtain:

$$\begin{aligned} f_j(z_1,z_2) &\le c_{j,0} + c_{0,j+1} + \min_{0\le\theta\le Q} f_j(\theta,Q-\theta) - c_{j,j+1} \\ &\le c_{j,0} + (c_{0,j} + c_{j,j+1}) - c_{j,j+1} + \min_{0\le\theta\le Q} f_j(\theta,Q-\theta) \qquad (6.16) \\ &\le c_{j,0} + c_{0,j} + \min_{0\le\theta\le Q} f_j(\theta,Q-\theta) \le 2c_{0,j} + \min_{0\le\theta\le Q} f_j(\theta,Q-\theta) \qquad \text{QED} \end{aligned}$$

Let now $H_j(z_1,z_2)$ and $H'_j(z_1,z_2)$ denote the values of part (a) and part (b) inside the minimisation of Eq. (6.10). Note that part (b) is independent of the values of z_1,z_2. We will show that a theorem similar to Theorem 1 (discussed in Section 6.2.2) holds for the unified load case.

Let z_1^*, z_2^* the item quantities on board after serving customer j.

THEOREM 2: *In the unified load case of Eq. (6.10), for each customer j, there exists a threshold function* $h^u{}_j(z_1, z_2) = c^u{}_j$*, such that the optimal decision, after serving customer j, is to continue to customer j+1 if* $h^u{}_j(z_1{}^*, z_2{}^*) \geq c^u{}_j$*, or return to the depot otherwise.*

The proof is similar to that of Theorem 1. We will show by induction that for all $(z_1,z_2) \in S_j$, $f_j(z_1,z_2)$ is a non-increasing function; that is $f_j(z_1+\delta_1, z_2+\delta_2) \leq f_j(z_1, z_2)$ for $z_1, z_2 \geq 0$. This relationship is true for the last customer n, where $f_n(z_1,z_2) = c_{n0}$ is independent of (z_1,z_2). Hence $f_n(z_1,z_2)$ is non-increasing with respect to $(z_1,z_2) \in S_n$. We will now prove that similarly to the compartmentalised case, if $f_{j+1}(z_1,z_2)$ is monotonically non-increasing with respect to $(z_1,z_2) \in S_{j+1}$, then $f_j(z_1,z_2)$ is also monotonically non-increasing with respect to $(z_1,z_2) \in S_j$.

Let:

$$H_j(z_1, z_2) = H_j^a(z_1,z_2) + H_j^b(z_1,z_2) + H_j^c(z_1,z_2) + H_j^d(z_1,z_2) = \tag{6.17}$$

$$= \left[c_{j,j+1} + \sum_{k_1:\xi^{k_1} \leq z_1} \;\; \sum_{k_2:\xi^{k_2} \leq z_2} f_{j+1}(z_1 - \xi^{k_1}, z_2 - \xi^{k_2}) p_{j+1}(k_1,k_2) \right]$$

$$+ \left[\sum_{k_1:z_1<\xi^{k_1}} \;\; \sum_{k_2:\xi^{k_2} \leq z_2} \left[2c_{j+1,0} + \min_{\xi^{k_1}-z_1 \leq \theta \leq Q} f_{j+1}(\theta - (\xi^{k_1} - z_1), Q-\theta) \right] p_{j+1}(k_1,k_2) \right]$$

$$+ \left[\sum_{k_1:\xi^{k_1} \leq z_1} \;\; \sum_{k_2:z_2<\xi^{k_2}} \left[2c_{j+1,0} + \min_{\xi^{k_2}-z_2 \leq \theta \leq Q} f_{j+1}(\theta, Q-\theta-(\xi^{k_2} - z_2)) \right] p_{j+1}(k_1,k_2) + \right]$$

$$+ \left[\sum_{k_1:z_1<\xi^{k_1}} \;\; \sum_{k_2:z_2<\xi^{k_2}} \left[2c_{j+1,0} + \min_{\xi^{k_1}-z_1 \leq \theta \leq Q-(\xi^{k_2}-z_2)} f_{j+1}(\theta - (\xi^{k_1} - z_1), Q-\theta-(\xi^{k_2} - z_2)) \right] p_{j+1}(k_1,k_2) \right]$$

We will calculate the difference $H_j(z_1, z_2) - H_j(z_1+\delta_1, z_2+\delta_2)$ by subtracting the terms that correspond to the same nine regions of Figures 6.7 and 6.8 for the first and second term of the difference respectively.

$$H_j(z_1, z_2) - H_j(z_1+\delta_1, z_2+\delta_2) = \tag{6.18}$$

$$\left[H_j^{a}(z_1,z_2)-H_j^{a'_1}(z_1+\delta_1,z_2+\delta_2)\right]+\left[H_j^{b_1}(z_1,z_2)-H_j^{a'_2}(z_1+\delta_1,z_2+\delta_2)\right]+\left[H_j^{b_2}(z_1,z_2)-H_j^{b'_1}(z_1+\delta_1,z_2+\delta_2)\right]+$$

$$+\left[H_j^{c_1}(z_1,z_2)-H_j^{a'_4}(z_1+\delta_1,z_2+\delta_2)\right]+\left[H_j^{d_1}(z_1,z_2)-H_j^{a'_3}(z_1+\delta_1,z_2+\delta_2)\right]+\left[H_j^{d_2}(z_1,z_2)-H_j^{b'_2}(z_1+\delta_1,z_2+\delta_2)\right]+$$

$$\left[H_j^{c_2}(z_1,z_2)-H_j^{c'_1}(z_1+\delta_1,z_2+\delta_2)\right]+\left[H_j^{d_4}(z_1,z_2)-H_j^{c'_2}(z_1+\delta_1,z_2+\delta_2)\right]+\left[H_j^{d_3}(z_1,z_2)-H_j^{d'}(z_1+\delta_1,z_2+\delta_2)\right]=$$

$$\left.\begin{aligned}=\Bigg[\, & c_{j,j+1}+\sum_{k_1:\xi^{k_1}\le z_1}\ \sum_{k_2:\xi^{k_2}\le z_2} f_{j+1}(z_1-\xi^{k_1},z_2-\xi^{k_2})\,p_{j+1}(k_1,k_2)\\ & -c_{j,j+1}-\sum_{k_1:\xi^{k_1}\le z_1}\ \sum_{k_2:\xi^{k_2}\le z_2} f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})\,p_{j+1}(k_1,k_2)\Bigg]\end{aligned}\right\}\ \text{term 1}$$

$$\left.\begin{aligned}+\Bigg[\, & \sum_{k_1:z_1<\xi^{k_1}\le z_1+\delta_1}\ \sum_{k_2:\xi^{k_2}\le z_2}\left[2c_{j+1,0}+\min_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}(\theta-(\xi^{k_1}-z_1),Q-\theta)\right]p_{j+1}(k_1,k_2)\\ & -\sum_{k_1:z_1<\xi^{k_1}\le z_1+\delta_1}\ \sum_{k_2:\xi^{k_2}\le z_2} f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})\,p_{j+1}(k_1,k_2)\Bigg]\end{aligned}\right\}\ \text{term 2}$$

$$\left.\begin{aligned}+\Bigg[\, & \sum_{k_1:z_1+\delta_1<\xi^{k_1}}\ \sum_{k_2:\xi^{k_2}\le z_2}\left[2c_{j+1,0}+\min_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}(\theta-(\xi^{k_1}-z_1),Q-\theta)\right]p_{j+1}(k_1,k_2)\\ & -\sum_{k_1:z_1<\xi^{k_1}+\delta_1}\ \sum_{k_2:\xi^{k_2}\le z_2}\left[2c_{j+1,0}+\min_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}(\theta-(\xi^{k_1}-z_1-\delta_1),Q-\theta)\right]p_{j+1}(k_1,k_2)\Bigg]\end{aligned}\right\}\ \text{term 3}$$

$$\left.\begin{aligned}+\Bigg[\, & \sum_{k_1:\xi^{k_1}\le z_1}\ \sum_{k_2:z_2<\xi^{k_2}\le z_2+\delta_2}\left[2c_{j+1,0}+\min_{\xi^{k_2}-z_2\le\theta\le Q} f_{j+1}(\theta,Q-\theta-(\xi^{k_2}-z_2))\right]p_{j+1}(k_1,k_2)\\ & -\sum_{k_1:\xi^{k_1}\le z_1}\ \sum_{k_2:z_2<\xi^{k_2}\le z_2+\delta_2} f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})\,p_{j+1}(k_1,k_2)\Bigg]\end{aligned}\right\}\ \text{term 4}$$

$$\left.\begin{aligned}+\Bigg[\, & \sum_{k_1:z_1<\xi^{k_1}\le z_1+\delta_1}\ \sum_{k_2:z_2<\xi^{k_2}\le z_2+\delta_2}\left[2c_{j+1,0}+\min_{\xi^{k_1}-z_1\le\theta\le Q-(\xi^{k_2}-z_2)} f_{j+1}(\theta-(\xi^{k_1}-z_1),Q-\theta-(\xi^{k_2}-z_2))\right]p_{j+1}(k_1,k_2)\\ & -\sum_{k_1:z_1<\xi^{k_1}\le z_1+\delta_1}\ \sum_{k_2:z_2<\xi^{k_2}\le z_2+\delta_2} f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})\,p_{j+1}(k_1,k_2)\Bigg]\end{aligned}\right\}\ \text{term 5}$$

$$\left.\begin{aligned}&+\Bigg[\sum_{k_1:z_1+\delta_1<\xi^{k_1}}\ \sum_{k_2:z_2<\xi^{k_2}\le z_2+\delta_2}\left[2c_{j+1,0}+\min_{\xi^{k_1}-z_1\le\theta\le Q-(\xi^{k_2}-z_2)} f_{j+1}(\theta-(\xi^{k_1}-z_1),Q-\theta-(\xi^{k_2}-z_2))\right]p_{j+1}(k_1,k_2)\\&\qquad-\sum_{k_1:z_1+\delta_1<\xi^{k_1}}\ \sum_{k_2:z_2<\xi^{k_2}\le z_2+\delta_2}\left[2c_{j+1,0}+\min_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}(\theta-(\xi^{k_1}-z_1-\delta_1),Q-\theta)\right]p_{j+1}(k_1,k_2)\Bigg]\end{aligned}\right\}\text{term 6}$$

$$\left.\begin{aligned}&+\Bigg[\sum_{k_1:\xi^{k_1}\le z_1}\ \sum_{k_2:z_2+\delta_2<\xi^{k_2}}\left[2c_{j+1,0}+\min_{\xi^{k_2}-z_2\le\theta\le Q} f_{j+1}(\theta,Q-\theta-(\xi^{k_2}-z_2))\right]p_{j+1}(k_1,k_2)\\&\qquad-\sum_{k_1:\xi^{k_1}\le z_1}\ \sum_{k_2:z_2+\delta_2<\xi^{k_2}}\left[2c_{j+1,0}+\min_{\xi^{k_2}-z_2\le\theta\le Q} f_{j+1}(\theta,Q-\theta-(\xi^{k_2}-z_2-\delta_2))\right]p_{j+1}(k_1,k_2)\Bigg]\end{aligned}\right\}\text{term 7}$$

$$\left.\begin{aligned}&+\Bigg[\sum_{k_1:z_1<\xi^{k_1}\le z_1+\delta_1}\ \sum_{k_2:z_2+\delta_2<\xi^{k_2}}\left[2c_{j+1,0}+\min_{\xi^{k_1}-z_1\le\theta\le Q-(\xi^{k_2}-z_2)} f_{j+1}(\theta-(\xi^{k_1}-z_1),Q-\theta-(\xi^{k_2}-z_2))\right]p_{j+1}(k_1,k_2)\\&\qquad-\sum_{k_1:z_1<\xi^{k_1}\le z_1+\delta_1}\ \sum_{k_2:z_2+\delta_2<\xi^{k_2}}\left[2c_{j+1,0}+\min_{\xi^{k_2}-z_2\le\theta\le Q} f_{j+1}(\theta,Q-\theta-(\xi^{k_2}-z_2-\delta_2))\right]p_{j+1}(k_1,k_2)\Bigg]\end{aligned}\right\}\text{term 8}$$

$$\left.\begin{aligned}&+\Bigg[\sum_{k_1:z_1+\delta_1<\xi^{k_1}}\ \sum_{k_2:z_2+\delta_2<\xi^{k_2}}\left[2c_{j+1,0}+\min_{\xi^{k_1}-z_1\le\theta\le Q-(\xi^{k_2}-z_2)} f_{j+1}(\theta-(\xi^{k_1}-z_1),Q-\theta-(\xi^{k_2}-z_2))\right]p_{j+1}(k_1,k_2)\\&-\sum_{k_1:z_1+\delta_1<\xi^{k_1}}\ \sum_{k_2:z_2+\delta_2<\xi^{k_2}}\left[2c_{j+1,0}+\min_{\xi^{k_1}-z_1\le\theta\le Q-(\xi^{k_2}-z_2)} f_{j+1}(\theta-(\xi^{k_1}-z_1-\delta_1),Q-\theta-(\xi^{k_2}-z_2-\delta_2))\right]p_{j+1}(k_1,k_2)\Bigg]\end{aligned}\right\}\text{term 9}$$

We can distinguish three types of terms in Eq.(6.18). The first type includes the first term. Considering this term we have:

$$H_j^{a}(\mathrm{z}_1,\mathrm{z}_2)-H_j^{a_1'}(\mathrm{z}_1+\delta_1,\mathrm{z}_2+\delta_2)= \tag{6.18.1}$$

$$=\sum_{\xi^{k_1}\le z_1}\ \sum_{\xi^{k_2}\le z_2}\left[f_{j+1}(z_1-\xi^{k_1},z_2-\xi^{k_2})-f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})\right]p_{j+1}(k_1,k_2)$$

Since $f_{j+1}(z_1, z_2)$ is monotonically non-increasing, this term is non-negative.

The second type of terms includes the third, sixth, seventh, eighth and nineth terms of Eq.(6.18). Considering the third term we have:

$$H_j^{b_2}(z_1,z_2)\ -\ H_j^{b'_1}(z_1+\delta_1,z_2+\delta_2)= \tag{6.18.2}$$

$$=\sum_{k_1:z_1+\delta_1<\xi^{k_1}}\ \sum_{k_2:\xi^{k_2}\le z_2}\left[2c_{j+1,0}+\min_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}(\theta-(\xi^{k_1}-z_1),Q-\theta)\right]p_{j+1}(k_1,k_2)$$

$$-\sum_{k_1:z_1+\delta_1<\xi^{k_1}}\ \sum_{k_2:\xi^{k_2}\le z_2}\left[2c_{j+1,0}+\min_{\xi^{k_1}-(z_1+\delta_1)\le\theta\le Q} f_{j+1}[\theta-[\xi^{k_1}-(z_1+\delta_1)],Q-\theta]\right]p_{j+1}(k_1,k_2)=$$

$$=\sum_{k_1:z_1+\delta_1<\xi^{k_1}}\ \sum_{k_2:\xi^{k_2}\le z_2}\left[\min_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}[\theta-(\xi^{k_1}-z_1),Q-\theta]-\min_{\xi^{k_1}-(z_1+\delta_1)\le\theta\le Q} f_{j+1}\left[\theta-[\xi^{k_1}-(z_1+\delta_1)],Q-\theta\right]\right]p_{j+1}(k_1,k_2)$$

Let $z'_1=\theta-(\xi^{k_1}-z_1)$ and $z'_2=Q-\theta$.

Then $f_{j+1}[\theta-(\xi^{k_1}-z_1),Q-\theta]=f_{j+1}(z'_1,z'_2)$ and $f_{j+1}\left[\theta-[\xi^{k_1}-(z_1+\delta_1)],Q-\theta\right]=f_{j+1}(z'_1+\delta_1,z'_2)$.

Since $f_{j+1}(z'_1,z'_2)$ is monotonically non-increasing, then $f_{j+1}(z'_1,z'_2)\ge f_{j+1}(z'_1+\delta_1,z'_2)$. Furthermore, by examining the minimization boundaries $\xi^{k_1}-z_1\le\theta\le Q$ and $\xi^{k_1}-(z_1+\delta_1)\le\theta\le Q$ it is clear that the first is a subset of the second.

Thus:

$$f_{j+1}(z'_1,z'_2)\ge f_{j+1}(z'_1+\delta_1,z'_2)\Leftrightarrow$$

$$\Leftrightarrow f_{j+1}[\theta-(\xi^{k_1}-z_1),Q-\theta]\ge f_{j+1}[\theta-(\xi^{k_1}-z_1-\delta_1),Q-\theta]\Leftrightarrow$$

$$\Leftrightarrow \min_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}[\theta-(\xi^{k_1}-z_1),Q-\theta]\ge \min_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}[\theta-(\xi^{k_1}-z_1-\delta_1),Q-\theta]\Leftrightarrow$$

$$\Leftrightarrow \min_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}[\theta-(\xi^{k_1}-z_1),Q-\theta]\ge \min_{\xi^{k_1}-z_1-\delta_1\le\theta\le Q} f_{j+1}[\theta-(\xi^{k_1}-z_1-\delta_1),Q-\theta]$$

Thus, $H_j^{b_2}(z_1,z_2)\ -\ H_j^{b'_1}(z_1+\delta_1,z_2+\delta_2)\ge 0$. Similarly we can show that the differences corresponding to the sixth, seventh, eighth and nineth terms are also non-negative.

The third type of terms includes the second, fourth, and fifth terms of Eq.(6.18). Considering the second term we have:

$$H_j^{b_1}(z_1,z_2)-H_j^{a'_2}(z_1+\delta_1,z_2+\delta_2)= \tag{6.18.3}$$

$$=\sum_{k_1:z_1<\xi^{k_1}\le z_1+\delta_1}\ \sum_{k_2:\xi^{k_2}\le z_2}\left[2c_{j+1,0}+\min_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}(\theta-(\xi^{k_1}-z_1),Q-\theta)-f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})\right]p_{j+1}(k_1,k_2)$$

If the following holds, then the above term is non-negative:

$$\begin{gathered}2c_{j+1,0}+\min_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}(\theta-(\xi^{k_1}-z_1),Q-\theta)-f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})\ge 0\Leftrightarrow\\ \Leftrightarrow f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})\le 2c_{j+1,0}+\min_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}(\theta-(\xi^{k_1}-z_1),Q-\theta)\end{gathered} \tag{6.18.4}$$

According to LEMMA 2 $f_{j+1}(z_1+\delta_1-\xi^{k_1},z_2+\delta_2-\xi^{k_2})\le 2c_{j+1,0}+\min_{0\le\theta\le Q} f_{j+1}(\theta,Q-\theta)$. (6.18.5)

However,

$$\min_{0\le\theta\le Q} f_{j+1}(\theta,Q-\theta)\le \min_{0\le\theta\le Q} f_{j+1}(\theta-(\xi^{k_1}-z_1),Q-\theta)\le \min_{\xi^{k_1}-z_1\le\theta\le Q} f_{j+1}(\theta-(\xi^{k_1}-z_1),Q-\theta)$$

The first inequality holds since f_{j+1} is non-decreasing (and $\xi^{k_1}>z_1$) and the second since the limits of θ are narrower in the last term. Thus, the inequality of Eq.(6.18.4) holds, and the difference of Eq.(6.18.3) is non-negative. Similarly we can show that the difference corresponding to fourth and fifth terms are also non-negative.

Assembling all the terms mentioned above we can clearly see that $H_j(z_1, z_2) - H_j(z_1+\delta_1, z_2+\delta_2)\ge 0$ and hence, $H_j(z_1, z_2)$ is a monotonically non-increasing function similarly to the respective equation in the compartmentalized load case. Note now that part (b) $H'_j(z_1,z_2)$ of Eq. (6.10) is independent of z_1,z_2 and thus a constant $H'_j(z_1,z_2)=H'_j$ in the z_1,z_2 space. Therefore $f_j(z_1,z_2)=\min\{H_j(z_1,z_2),H'_j(z_1,z_2)\}$ is a non-increasing function.

Furthermore, the intersection $H_j(z_1,z_2) \cap H'_j$ can in general be described by a function of the form $h_j(z_1,z_2)=c$. This is more clearly shown in Figure 6.24. Every (z_1,z_2) combination that lies within the highlighted area (red points), $h_j(z_1,z_2)<c_j$, corresponds to a depot return before visiting customer $j+1$. For (z_1,z_2) combinations that lie outside the highlighted area (green points) $h_j(z_1,z_2)\geq c_j$, the vehicle should proceed to the next customer.

This concludes the proof of Theorem 2, which, similarly to Theorem 1, provides the optimal routing policy; i.e. the policy that if followed by the vehicle, we will obtain the minimum expected value of the travel cost.

6.3.3 Solution Algorithm

Similarly to the compartmentalized multiple product case, and based on the formulation presented in Section 6.3.1, we developed an appropriate algorithm that uses Dynamic Programming to derive the optimal solution of the unified load problem in a reasonable amount of time. Consider the 5-customer network of Figure 6.23. The total vehicle capacity (for both products) is $Q=10$ units; the demand ξ_j^i for each product i (i = 1,2) per customer j (j = 1, …, 5) is given in Appendix D, and the distances between the nodes c_{ij} are given in Figure 6.23.

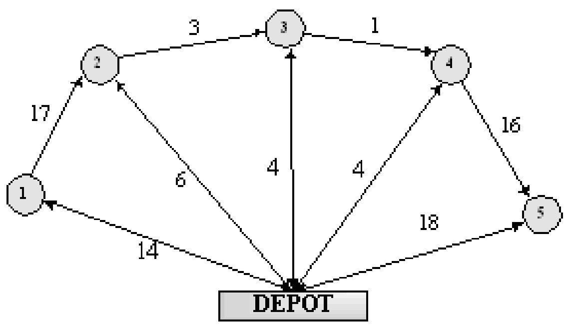


Figure 6.23. 5-customer network for the unified load case.

The problem is solved using the dynamic programming algorithm presented in Section 6.3.1. Let (z_1, z_2) be the remaining quantities in the vehicle after customer j has been served. Let $f_j(z_1, z_2)$ and $x_j(z_1, z_2)$ be the minimum expected cost and the corresponding decision after customer $j \in \{1, 2, 3, 4, 5\}$ has been served. Clearly, $f_5(z_1, z_2) = 18, x_5(z_1, z_2) = 1$ for $z_1, z_2 \in \{0,\ldots,10\}$, $z_1+z_2 \leq 10$.

In Tables 6.6-6.10 we provide the results for nodes 0, 1, 2, 3, 4. In these Tables, (z_1, z_2) represents the quantity carried by the vehicle after customer j has been served; each cell includes two values: The first is the value of $x_j(z_1, z_2)$ and the second is the value of $f_j(z_1, z_2)$.

Table 6.6. Results obtained for node 4.

z2 \ z1	0	1	2	3	4	5	6	7	8	9	10
0	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1
1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	-
2	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	-	-
3	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	1; 41,1	-	-	-
4	1; 41,1	1; 41,1	0; 35,8	0; 35,6	0; 35,5	0; 35,3	0; 35,1	-	-	-	-
5	1; 41,1	1; 41,1	0; 35,7	0; 35,5	0; 35,3	0; 35,1	-	-	-	-	-
6	1; 41,1	1; 41,1	0; 35,6	0; 35,4	0; 35,1	-	-	-	-	-	-
7	1; 41,1	1; 41,1	0; 35,5	0; 35,2	-	-	-	-	-	-	-
8	1; 41,1	1; 41,1	0; 35,4	-	-	-	-	-	-	-	-
9	1; 41,1	1; 41,1	-	-	-	-	-	-	-	-	-
10	1; 41,1	-	-	-	-	-	-	-	-	-	-

Table 6.7. Results obtained for node 3.

z1 / z2	0	1	2	3	4	5	6	7	8	9	10
0	1; 44,6	0; 44,5	0; 44,4	0; 44,4	0; 44,4	0; 44,3	0; 44,3	0; 44,3	0; 44,3	0; 44,3	0; 44,3
1	1; 44,6	0; 43,1	0; 43,0	0; 42,8	0; 42,7	0; 42,7	0; 42,7	0; 42,7	0; 42,7	0; 42,7	-
2	0; 44,5	0; 43,1	0; 42,5	0; 42,3	0; 42,2	0; 42,2	0; 42,2	0; 42,2	0; 42,2	-	-
3	0; 44,5	0; 43,0	0; 42,5	0; 42,2	0; 42,2	0; 42,2	0; 42,2	0; 42,2	-	-	-
4	0; 44,5	0; 43,0	0; 42,5	0; 42,2	0; 42,1	0; 42,1	0; 42,1	-	-	-	-
5	0; 44,5	0; 43,0	0; 42,5	0; 39,0	0; 38,8	0; 38,1	-	-	-	-	-
6	0; 44,5	0; 43,0	0; 42,4	0; 38,9	0; 37,6	-	-	-	-	-	-
7	0; 44,5	0; 43,0	0; 42,4	0; 38,7	-	-	-	-	-	-	-
8	0; 44,5	0; 43,0	0; 42,4	-	-	-	-	-	-	-	-
9	0; 44,5	0; 43,0	-	-	-	-	-	-	-	-	-
10	0; 44,5	-	-	-	-	-	-	-	-	-	-

Table 6.8. Results obtained for node 2.

z1 / z2	0	1	2	3	4	5	6	7	8	9	10
0	1; 52,4	1; 52,4	1; 52,4	1; 52,4	1; 52,4	1; 52,4	1; 52,4	1; 52,4	1; 52,4	1; 52,4	1; 52,4
1	1; 52,4	1; 52,4	0; 52,3	0; 52,2	0; 52,2	0; 52,2	0; 52,2	0; 52,1	0; 52,1	0; 52,1	-
2	1; 52,4	0; 50,1	0; 50,0	0; 49,9	0; 49,9	0; 49,8	0; 49,8	0; 49,8	0; 49,8	-	-
3	0; 50,1	0; 49,9	0; 49,3	0; 49,3	0; 49,2	0; 49,1	0; 49,1	0; 49,1	-	-	-
4	0; 50,0	0; 48,3	0; 47,7	0; 47,3	0; 47,2	0; 47,1	0; 47,0	-	-	-	-
5	0; 50,0	0; 48,3	0; 46,7	0; 46,2	0; 45,9	0; 45,8	-	-	-	-	-
6	0; 50,0	0; 48,2	0; 46,6	0; 45,9	0; 45,5	-	-	-	-	-	-
7	0; 50,0	0; 48,2	0; 46,6	0; 45,8	-	-	-	-	-	-	-
8	0; 50,0	0; 48,2	0; 46,6	-	-	-	-	-	-	-	-
9	0; 50,0	0; 48,2	-	-	-	-	-	-	-	-	-
10	0; 50,0	-	-	-	-	-	-	-	-	-	-

Table 6.9. Results obtained for node 1.

z2 \ z1	0	1	2	3	4	5	6	7	8	9	10
0	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6
1	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	-
2	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	-	
3	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	1; 67,6	-	-	
4	1; 67,6	1; 67,6	1; 67,6	0; 67,4	0; 67,2	0; 67,1	0; 67,0	-	-	-	-
5	1; 67,6	1; 67,6	0; 67,3	0; 66,5	0; 66,3	0; 66,1	-			-	-
6	1; 67,6	1; 67,6	0; 66,3	0; 65,4	0; 64,6	-	-			-	-
7	1; 67,6	1; 67,6	0; 66,3	0; 64,7	-	-	-	-	-	-	-
8	1; 67,6	1; 67,6	0; 66,2	-	-	-	-	-	-	-	-
9	1; 67,6	1; 67,6	-	-	-	-	-	-	-	-	-
10	1; 67,6	-	-	-	-	-	-	-	-	-	-

$H_1 > H'_1$

$H_1 \leq H'_1$

Table 6.10. Results obtained for node 0 (depot).

z2 \ z1	0	1	2	3	4	5	6	7	8	9	10
0	109,5	109	108,7	108,7	108,6	108,6	108,6	108,6	108,5	108,5	108,5
1	109,2	107,8	107,3	107,2	107,2	107,1	107	107	106,9	106,9	-
2	108,3	106,6	101,1	101,1	101	100,9	100,8	100,7	100,6	-	-
3	107,6	88,5	83,1	82,9	82,8	82,7	82,6	82,5	-	-	-
4	107,5	88,5	83	82,8	82,7	82,5	82,3	-	-	-	-
5	107,5	88,4	82,9	82,7	82,5	**82,4**	-	-	-	-	-
6	107,5	88,4	82,8	82,6	82,5	-	-	-	-	-	-
7	107,4	88,3	82,7	82,5	-	-	-	-	-	-	-
8	107,4	88,3	82,6	-	-	-	-	-	-	-	-
9	107,4	88,2	-	-	-	-	-	-	-	-	-
10	107,4	-	-	-	-	-	-	-	-	-	-

Therefore the minimum expected total cost is equal to 82,4 for $z_1 = 5$ and $z_2 = 5$.

Figure 6.24 illustrates the threshold function $h_1(z_1, z_2) = c$ as obtained from the results of Table 6.9. It is clearly seen from this table that $H'_1 = 67{,}6$. For all (z_1, z_2) pairs with $H_1(z_1, z_2) \leq H'_1$ the vehicle should proceed to the next customer directly.

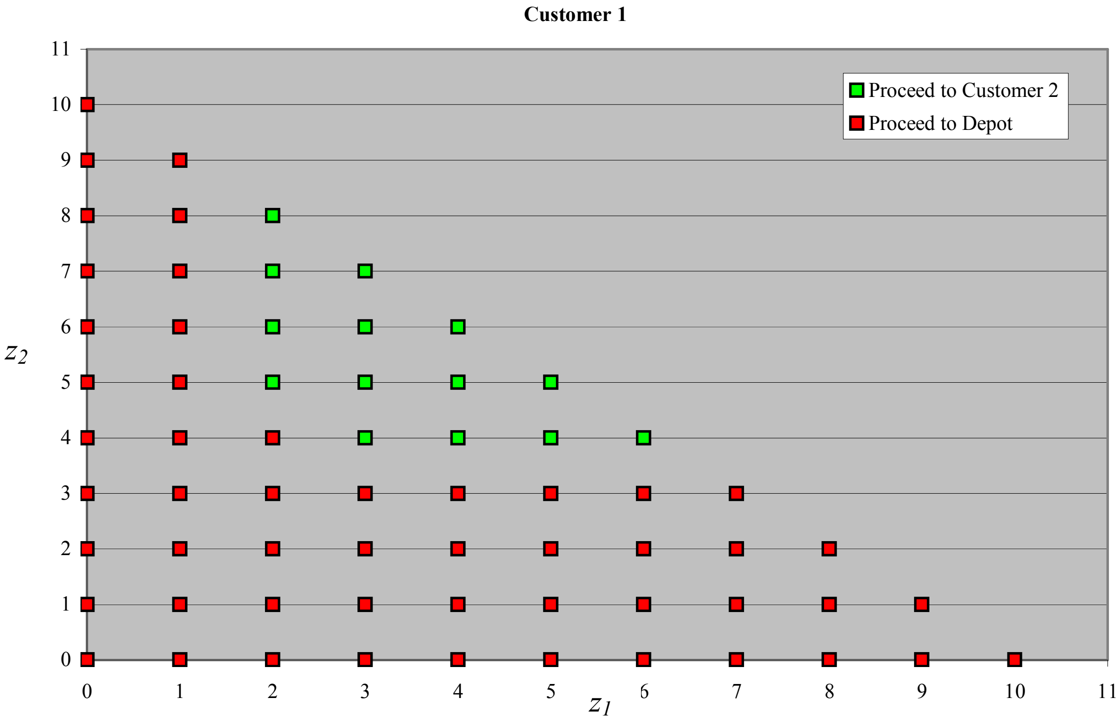


Figure 6.24. The corresponding load combinations after serving customer 1.

The performance of the algorithm was also found to be within acceptable levels. As an indication, for a test-problem of 10 customer points, 2 products and total vehicle capacity of 10 units, the algorithm derived the minimum expected cost, as well as the threshold curves per customer within 1420 seconds (=23,6 min). The number of combinations examined for each of the customer points 2-9 were approximately 8600. One needs to take into consideration that this problem is significantly more complex than its compartmentalized counterpart (approximately 1330 combinations per customer) due to the additional minimization steps required to identify the optimal θ and s values for each customer point. The experiments were run on a PC equipped with Intel Pentium IV, at 2.4 GHz and 512 MB of RAM.

In order to further assess the performance of the algorithm, a large number of problem test cases were created and run (see Figure 6.25). Three different problem test cases are plotted this figure. The first test case (10,000 randomly generated problems) concerns a vehicle with a total capacity (sum for both products) Q = 5 and is shown in green color. Each one of the 10 points of the curve is the average of 1,000 randomly generated problems. The second test case (again 10,000 randomly generated problems) concerns a vehicle with a total capacity Q = 10 and is shown in blue color. The third test case (10,000 randomly generated problems) included a vehicle with a total capacity Q = 15, and is shown in red color. The demand distributions were generated randomly (uniform distributions).

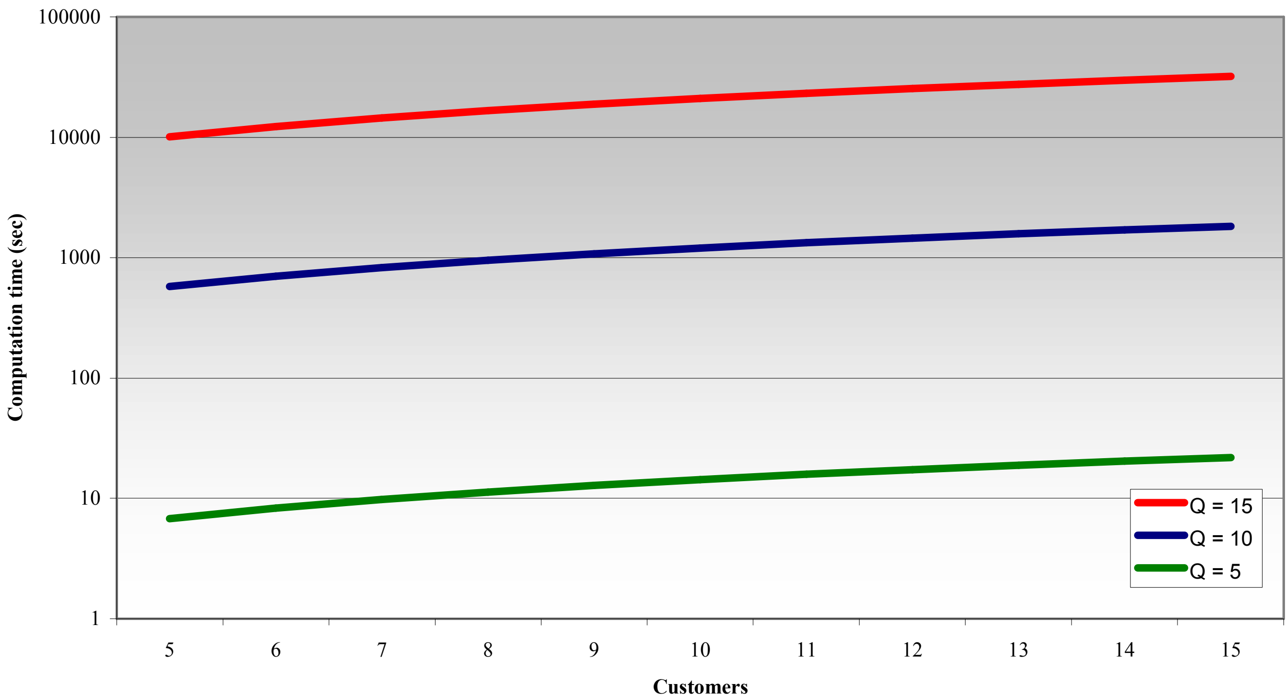


Figure 6.25. Performance results of the algorithm for the unified load case.

The trends obtained are similar to the ones corresponding to the compartmentalized case. For example, from Figure 6.25 it can be clearly seen that for a given number of customers, the increase of the vehicle capacity resulted in an almost exponential increase in the computation time of the algorithm (note that the scale of the y-axis of the graph is on a logarithmic scale).

On the other hand, if the capacity of the vehicle is kept constant, and the number of customers is increased, the computation time also increases. In order to analyze this trend further, ten different test cases were run with Q=10, ranging from five to fifty customer points (1000 randomly generated problems for each customer set) as shown in Figure 6.26. Each point in this figure represents the average time for the 1000 problems of the particular problem set. Figure 6.26 indicates that the increase in the computation time is linear with the number of customers.

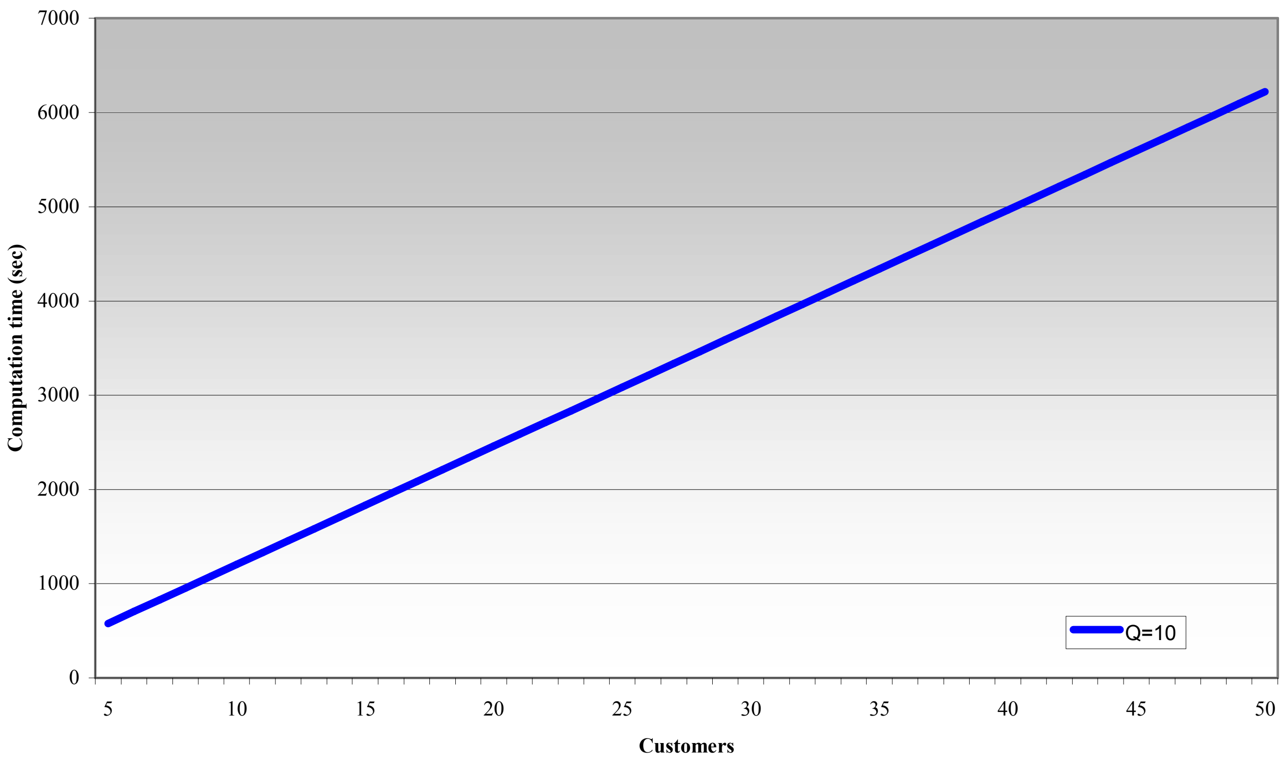


Figure 6.26. Performance results with up to 50 customer points.

For the 50-customer instance, the algorithm took approximately 6220 seconds (= 104 minutes) in order to obtain the solution of one problem.

6.4 Conclusions

This chapter focused on the multiple products extension of the Stochastic version of the Vehicle Routing with Depot Returns Problem (SVRDRP). This extension comprised of two cases; the compartmentalized and the unified load case. The objective of both these cases was to serve all customers with a single vehicle and minimize the expected value of the travel cost.

For both cases we presented the characteristics of each problem, a method to determine the minimum expected cost, and the theoretical results that permit us to determine the optimal decision after serving each customer. Both cases were formulated through dynamic programming, and for both it was proven that there exists an appropriate threshold function for each customer that distinguishes two regions in the space of possible loads after serving the customer: the region of loads for which the optimal decision (after serving the customer) is to return to the depot, and the region for which the optimal decision is to continue to the next customer. Both cases were formulated and solved for two products, but the results can be extended to n products (see Appendix B for the 3-product formulation of the compartmentalized case).

For both cases, through the execution of a large number of randomly generated problems it was found that the increase of the vehicle capacity results in an almost exponential increase in the computational time of the algorithm. On the other hand, if the capacity of the vehicle is kept constant, and the number of customers is increased, the computational time increases linearly with the number of customers. The unified load case proved to be significantly more complex than the compartmentalized load one, as expected. This is mainly due to the increased complexity of the formulation, which includes additional minimization steps in order to identify the replenishment stock for each customer point.

Chapter 7

Pickup and Delivery under Random Demand

7.1 Introduction

In this Chapter we examine the Pickup and Delivery case of the VRDRP under random demand. The characteristics and the mathematical formulation of the problem are described first. Subsequently, a new algorithm that solves this problem to optimality is presented. Finally, the performance of the algorithm is analyzed by solving a large number of sample problems.

7.2 The Pickup and Delivery SVRDRP

This case addresses an existing business model, according to which a vehicle can both sell (deliver) products but also collect (pickup) items from its customer base. An example is the distribution of paper-rolls for newspapers, in which, the paper-rolls are delivered to the customer on pallets and empty pallets are collected from the customer to be returned to the depot.

The demand of each customer (for either delivery or pickup) is again not known in advance, and it is revealed when the vehicle arrives at the customer site. However, as before, the sequence of serving the customers is predefined and the distances among all points in the network (depot and customer sites) are known. Upon completion of service at customer site j, the vehicle's driver has to make the same decision as the one described in Sections 5.2, 6.2 and 6.3, i.e. proceed to customer $j+1$, or return to the depot in order to empty the items that were picked up, refill the vehicle, and resume the route to serve customer $j+1$. An additional decision should be made concerning the quantity to be loaded to the vehicle each time the vehicle returns to the depot. This is because unnecessarily high stock levels may prevent the collection of returned items, therefore causing additional depot returns and lower customer service.

It is noted that the vehicle may have to visit a customer twice (but not more), if it cannot fully meet the demand of this customer during the first visit (for either delivery or pickup). It is assumed that service at a customer site and loading/unloading at the depot are performed instantly upon arrival of the vehicle. As in the multiple product case, the objective of the problem is to serve all customers and minimize travel cost in an expected value sense.

7.3 Dynamic Programming Formulation

Let z represent the product quantity and b the quantity of the returned items on board after serving customer j. Let ξ_j be the stochastic product demand (to be delivered to) customer $j \in \{1, \ldots, n\}$ and $p_{jk} = \mathrm{P}(\xi_j = \xi^k)$ its probability mass function. Also, let ρ_j be the stochastic demand for the items to be picked up from customer $j \in \{1, \ldots, n\}$ and $\pi_{jm} = \mathrm{P}(\rho_j = \rho^m)$ its probability mass function. Note that ξ_j and ρ_j are independent, and neither may exceed the vehicle capacity Q. Finally let $f_j(z,b)$ denote the minimum expected cost from customer j onward. Note that the quantities a) of the delivered product and b) of the picked up (returned) items are measured using the same unit of measure, e.g. m^3 or kg.

The mathematical formulation for minimizing the expected value of the route cost is given below:

$$f_j(z,b) = \min \tag{7.1}$$

$$\left\{\begin{array}{l}
c_{j,j+1} + \sum\limits_{k:\xi^k \le z} \sum\limits_{m:\rho^m + b \le Q-(z-\xi^k)} f_{j+1}(z-\xi^k, b+\rho^m) p_{j+1,k}\pi_{j+1,m} \\
\quad + \sum\limits_{k:z<\xi^k} \sum\limits_{m:\rho^m + b \le Q} \left[2c_{j+1,0} + \min\limits_{\xi^k - z \le \theta \le Q} f_{j+1}(\theta-(\xi^k - z),0)\right] p_{j+1,k}\pi_{j+1,m} \\
\quad + \sum\limits_{k:\xi^k \le z} \sum\limits_{m:Q-(z-\xi^k)<\rho^m+b} \left[2c_{j+1,0} + \min\limits_{0 \le \theta \le Q-[\rho^m - Q + (z-\xi^k)+b]} f_{j+1}[\theta, \rho^m - [Q-(z-\xi^k)-b]]\right] p_{j+1,k}\pi_{j+1,m} \\
+ \sum\limits_{k:z<\xi^k} \sum\limits_{m:Q<\rho^m+b} \left[2c_{j+1,0} + \min\limits_{\xi^k - z \le \theta \le 2Q - z + \xi^k - \rho^m - b} f_{j+1}[\theta-(\xi^k - z), [\rho^m - (Q-b)]]\right] p_{j+1,k}\pi_{j+1,m} \\
\\
\min\limits_{0 \le \theta \le Q} \left\{\begin{array}{l}
c_{j,0} + c_{0,j+1} \\
\quad + \sum\limits_{k:\xi^k \le \theta} \sum\limits_{m:\rho^m + \theta - \xi^k \le Q} f_{j+1}(\theta - \xi^k, \rho^m) p_{j+1,k}\pi_{j+1,m} \\
+ \sum\limits_{k:\theta<\xi^k} \sum\limits_{m:\rho^m \le Q} \left[2c_{j+1,0} + \min\limits_{\xi^k - \theta \le s \le Q} f_{j+1}(s-(\xi^k - \theta),0)\right] p_{j+1,k}\pi_{j+1,m} \\
\quad + \sum\limits_{k:\xi^k \le \theta} \sum\limits_{m:Q-(\theta-\xi^k)<\rho^m} \left[2c_{j+1,0} + \min\limits_{0 \le s \le Q-[\rho^m - Q + (\theta-\xi^k)]} f_{j+1}[s, \rho^m - [Q-(\theta-\xi^k)]]\right] p_{j+1,k}\pi_{j+1,m}
\end{array}\right\}
\end{array}\right.$$

Note that the second part of Eq. (7.1) does not contain a fifth term, since ρ^m cannot exceed the vehicle capacity Q.

In this case there are additional issues to be considered. First, upon visiting the depot, an additional decision needs to be made regarding the quantity of stock to be loaded onto the vehicle. This is due to the fact that it may not prove cost effective to load the vehicle to its full capacity, as there needs to be some space available in order to be able to accommodate the returned items as well without causing unnecessary depot returns. Let θ be the product quantity to be loaded to the vehicle during its (first) depot return. Then the space left for the product to be picked up will be $Q-\theta$. If the depot return comes after serving client $j+1$ (i.e. the first part of Eq. (7.1)) then the quantity of product loaded onto the vehicle can be such that the demand of client $j+1$ can be fully satisfied (either for delivery product or for returned items). However if the depot return comes before visiting client $j+1$ (i.e. second part of Eq. (7.1)) then another (subsequent) return to the depot will be necessary in case the product quantity θ loaded onto the vehicle (or space $Q-\theta$ left for the items to be picked up) is not adequate to fully satisfy the client's demand. During the second return, the decision to load stock s of the product to be delivered is an informed one, since the demand of customer $j+1$ is fully known.

The first part of the minimization equation consists of four distinct terms. Each term corresponds to one of the four areas shown in Figure 7.1. In this Figure the x-axis represents the stock on board, and the y-axis represents the space available in the vehicle after delivery has occurred.

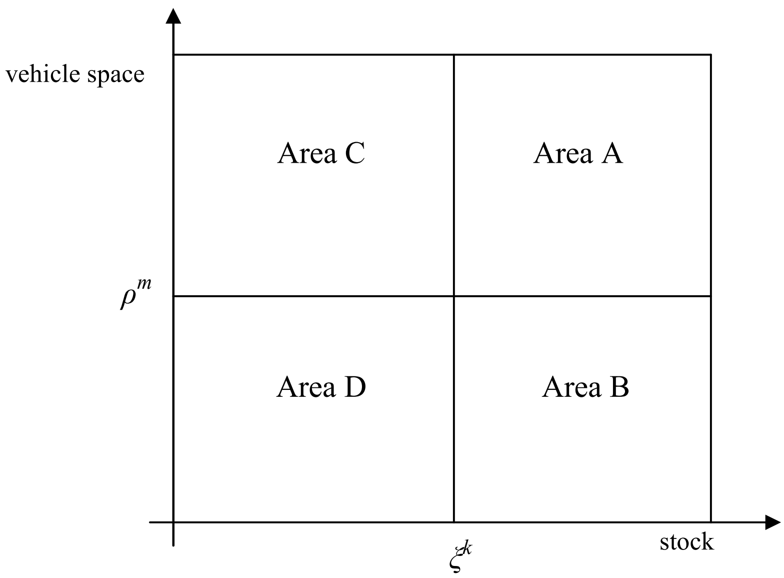


Figure 7.1. The solution space per customer point.

The first term of Eq. (7.1) corresponds to Area A and represents the cost incurred if the vehicle proceeds to the next customer directly and a) the stock z of the product is sufficient to satisfy the demand ($\xi^k \leq z$), and

b) the space left for the items to be picked up is also adequate ($\rho^m + b \leq Q - (z - \xi^k)$) (Area A in Figure 7.1). This case is shown in Figure 7.2.

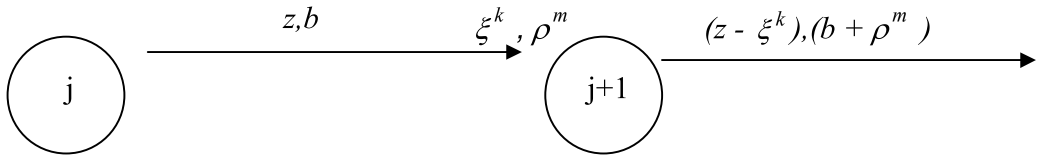


Figure 7.2. Both the stock and the vehicle space are sufficient to satisfy the demand ξ^k and ρ^m .

The second term of Eq. (7.1) corresponds to Area C of Figure 7.1 and represents the cost incurred if the vehicle goes to the next customer directly and the stock on board is not sufficient to fully satisfy the demand of the next customer ($\xi^k > z$), while the space left for the items to be picked up is sufficient to satisfy the demand for the returned items ($\rho^m + b \leq Q$). This case is represented in Figure 7.3 below.

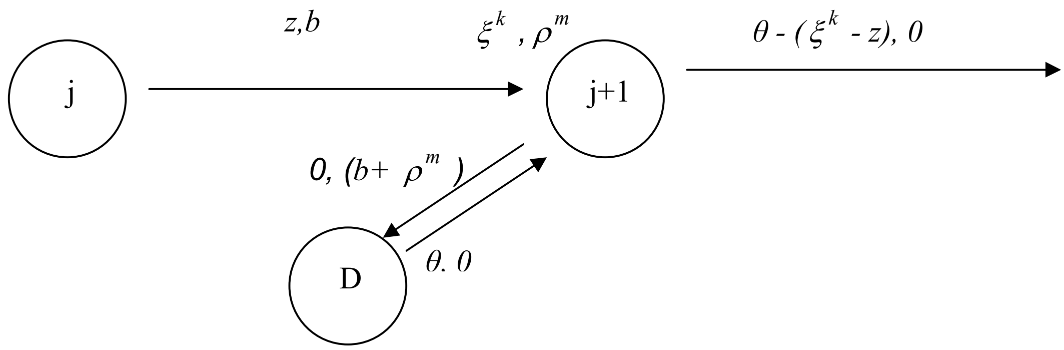


Figure 7.3. The stock is not sufficient but the vehicle space is.

The third term of Eq. (7.1) corresponds to Area B of Figure 7.1 and represents the cost incurred if the vehicle goes to the next customer directly and the space left for the items to be picked up is not sufficient to fully satisfy the demand of the next customer ($\rho^m + b > Q - (z - \xi^k)$), while the product quantity on board is sufficient to satisfy the corresponding demand of the next customer ($\xi^k \leq z$). This case is schematically represented in Figure 7.4.

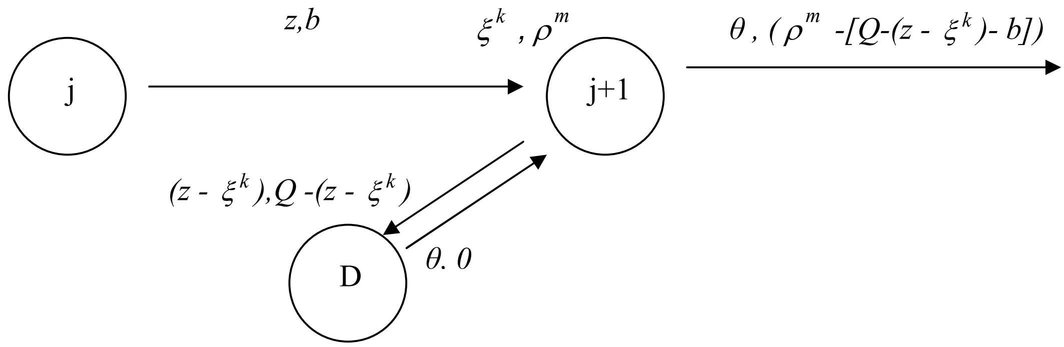


Figure 7.4. The stock is sufficient but the vehicle space is not.

The fourth term of Eq. (7.1) corresponds to Area D of Fig.(7.1) and represents the cost incurred if the vehicle goes to the next customer directly and both the product quantity on board as well as the space left for the items to be picked up are not sufficient to fully satisfy the demand of the next customer ($\xi^k > z$, $\rho^m + b >$ Q). This case is shown in Figure 7.5.

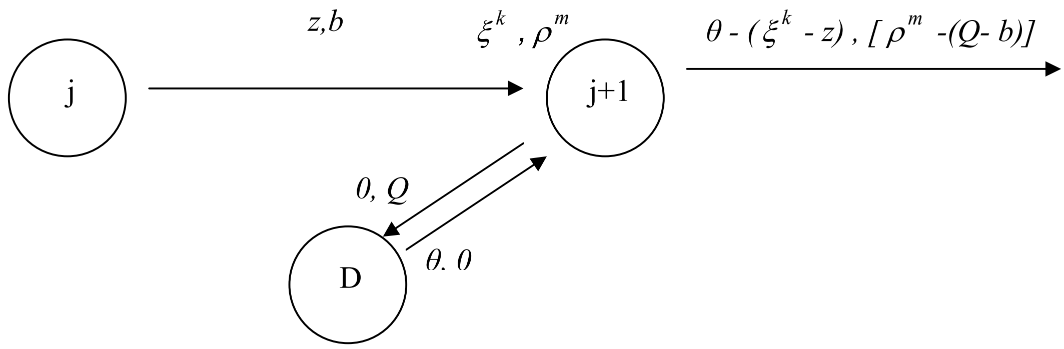


Figure 7.5. Neither the stock nor the vehicle space are sufficient.

The second part of the minimization equation consists of three terms and represents all possible cases that may occur when a proactive depot return is performed. In this case the values of *z* and *b* do not affect the result. Furthermore, since ξ^k and ρ^m are not known, the quantity θ (and the space available Q-θ) loaded at the depot may not be sufficient to satisfy the demand of customer *j+1*. Thus, a second, subsequent visit to the depot may be necessary. The first term of the second part represents the case in which the vehicle visits the depot, loads a quantity θ of the product (thus the vehicle space left is Q-θ), proceeds to customer $j+1$ and can fully satisfy its demand for both delivery ($\xi^k \leq \theta$) and pickup ($\rho^m \leq Q-\theta$). This is shown in Figure 7.6.

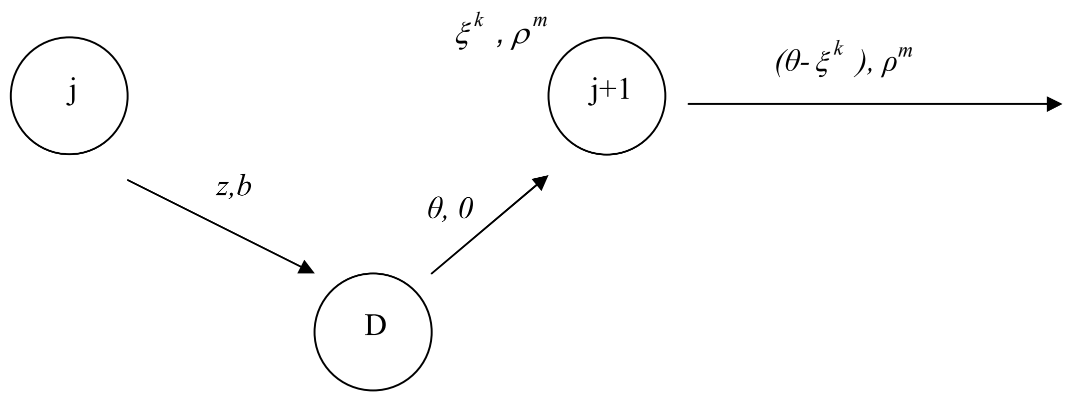


Figure 7.6. The case in which $\theta \geq \xi^k$ and Q- $\theta \geq \rho^m$.

The second term represents the case in which after visiting customer $j+1$ the vehicle cannot satisfy the customer's demand for delivery ($\xi^k > \theta$). In this case, the vehicle will deliver its entire load θ and pickup the entire quantity ρ^m (since $\rho^m \leq Q$ and the vehicle is empty after delivery). Subsequently, it will return to the depot once more, and make an additional, informed this time, decision for load quantity s for the product to be delivered in order to guarantee that it will fully satisfy the demand of customer $j+1$. This case is shown in Figure 7.7.

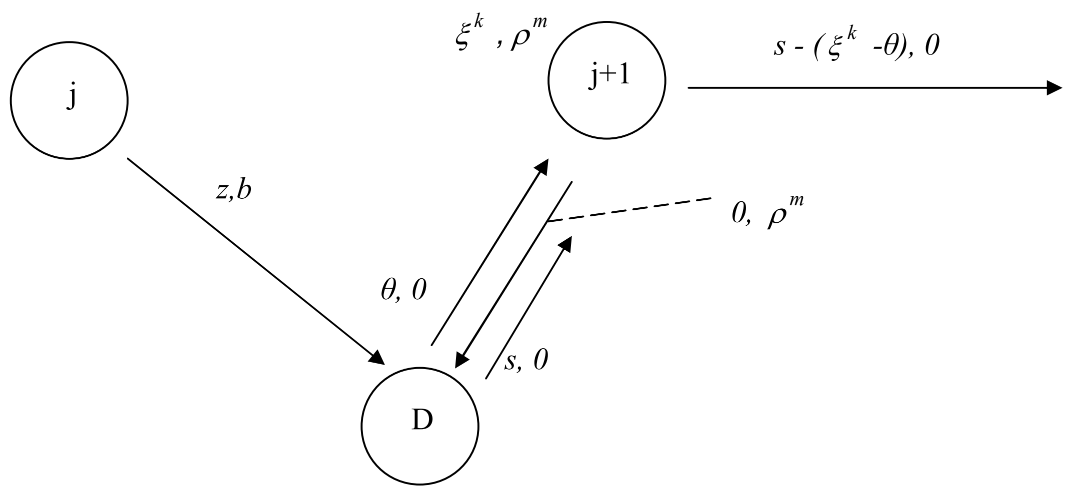


Figure 7.7. The case that $\theta < \xi^k$.

The third term represents the case in which after visiting customer $j+1$ the vehicle cannot satisfy the customer's demand for the product to be picked up ($\rho^m > Q-(\theta-\xi^k)$). The course of action here is analogous to the one discussed previously. This case is shown in Figure 7.8.

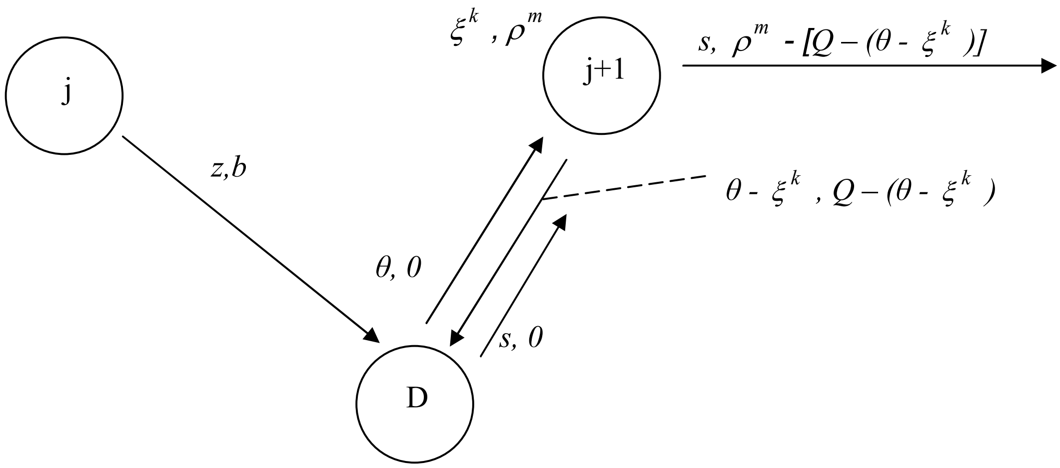


Figure 7.8. The case that $\rho^m > Q\text{-}\theta$.

Note that a fourth term is not present in this part of the equation, since when $\theta < \xi^k$ the vehicle delivers its entire load θ and thus, can pickup the entire quantity $\rho^m \le Q$.

7.4 Solution Algorithm

Similarly to the multiple product case presented in Chapter 6, and based on the formulation presented in Section 7.3, we developed an appropriate algorithm that uses Dynamic Programming to derive the optimal solution of the pickup and delivery problem in a reasonable amount of time.

The solution algorithm proceeds as follows: For each combination z and b and for each step of the algorithm, both terms of Eq. (7.1) are calculated and the one with the smallest value is selected. For calculating each term of Eq. (7.1) all allowable values of θ and s are tested and the appropriate minima are selected.

As an illustrative example, consider the 5-customer network of Figure 7.9. The vehicle capacity is $Q = 5$; the demand for delivery ξ^k and pickup ρ^m for each customer j (j = 1,…, 5) are given in Appendix E, and the distances between the nodes c_{ij} are given in Figure 7.9.

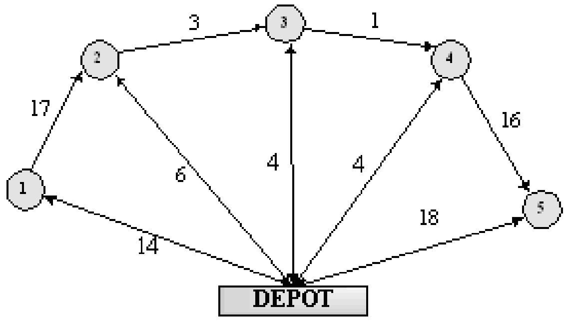


Figure 7.9. 5-customer network for pickup and delivery extension.

The problem is solved using the dynamic programming algorithm presented in Section 7.3. Let $f_j(z,b)$ and $x_j(z,b)$ be the minimum total cost and the corresponding decision after customer $j \in \{1,2,3,4,5\}$ has been served. The remaining quantity in the vehicle is (z,b). Clearly, $f_5(z_5,b_5) = 18, x_5(z_5,b_5) = 1$ for $0 \le z, b \le 5$.

In Tables 7.1-7.4 we provide the results for nodes 1, 2, 3, 4. In these Tables, the values z and b represent the quantities carried by the vehicle for the delivery product and the returned item after customer j has been served; each cell includes two values: The first is the value of $x_j(z,b)$ and the second is the value of $f_j(z,b)$. Furthermore, if we call $H_j(z,b)$ the first part of Eq. (7.1) and $H'_j(z,b)$ the second part of this Equation, the results of Tables 7.1 – 7.4 indicate which term is the minimum for each combination (z,b).

Table 7.1. Results obtained for node 4.

z_4 \ b_4	0	1	2	3	4	5
0	1; 44,18	1; 44,18	1; 44,18	1; 44,18	0; 35,59	0; 38,18
1	1; 44,18	1; 44,18	1; 44,18	1; 44,18	0; 38,54	-
2	1; 44,18	1; 44,18	1; 44,18	1; 44,18	-	-
3	1; 44,18	1; 44,18	1; 44,18	-	-	-
4	1; 44,18	1; 44,18	-	-	-	-
5	1; 44,18	-	-	-	-	-

$H_4 > H'_4$

Table 7.2. Results obtained for node 3.

$H'_3 = 51{,}23$ – no depot return

z_3 \ b_3	0	1	2	3	4	5
0	0; 46,23	0; 45,84	0; 45,49	0; 45,47	0; 46,15	0; 45,27
1	0; 46,25	0; 45,86	0; 45,70	0; 46,25	0; 47,59	-
2	0; 46,49	0; 46,10	0; 46,49	0; 47,67	-	-
3	0; 47,33	0; 46,96	0; 47,94	-	-	-
4	0; 48,72	0; 48,43	-	-	-	-
5	0; 50,83	-	-	-	-	-

$H_3 < H'_3$

Table 7.3. Results obtained for node 2.

z_2 \ b_2	0	1	2	3	4	5
0	1; 56,38	0; 56,21	0; 55,83	0; 54,43	0; 49,51	0; 49,39
1	1; 56,38	0; 56,23	0; 55,88	0; 54,61	0; 50,15	-
2	1; 56,38	0; 56,28	0; 56,00	0; 55,00	-	-
3	1; 56,38	0; 56,35	0; 56,22	-	-	-
4	1; 56,38	1; 56,38	-	-	-	-
5	1; 56,38	-	-	-	-	-

$H_2 > H'_2$

Table 7.4. Results obtained for node 1.

b_1 \ z_1	**0**	**1**	**2**	**3**	**4**	**5**
0	1; 75,98	1; 75,98	0; 75,80	0; 73,90	0; 73,56	0; 73,48
1	1; 75,98	1; 75,98	0; 74,96	0; 74,48	0; 74,82	-
2	1; 75,98	1; 75,98	0; 75,51	0; 75,60	-	-
3	1; 75,98	1; 75,98	1; 75,98	-	-	-
4	1; 75,98	1; 75,98	-	-	-	-
5	1; 75,98	-	-	-	-	-

$H_1 > H'_1$

Table 7.5. Results obtained for node 0 (depot).

b_0 \ z_0	**0**	**1**	**2**	**3**	**4**	**5**
0	116,43	113,87	111,32	92,73	**91,10**	90,51

Therefore the minimum expected total cost is equal to 91,10. For these tables, it is clear that we cannot draw a conclusion regarding the monotonicity of the function $f_j(z,b)$ similar to the conclusion drawn for the cases of Chapter 6. For example, the first row of Table 7.1 indicates that there is no definite (decreasing) trend as the value of z increases for $b=0$. Also, in Table 7.4, for $z=2$, there is no definite (decreasing) trend regarding the value of b.

Figure 7.10 illustrates the area $(z_3,b_3)\in S_3$ of customer-3 in a richer example for the same method where Q=10. As before, the algorithm calculates the value of the first part of Eq. (7.1) for all the (z,b) combinations. For each combination, the value obtained is compared with the constant value of the second part. If the value of the first part is found to be less than the value of the second part, it would mean that for this particular (z,b) combination the vehicle should proceed to the next customer directly (values shown in Figure 7.10 as green square points). If the value of the first part is found to be greater than the value of the second part, it would mean that for this particular (z,b) combination the vehicle should proceed to the next customer via the depot (values shown in Figure 7.10 as red square points).

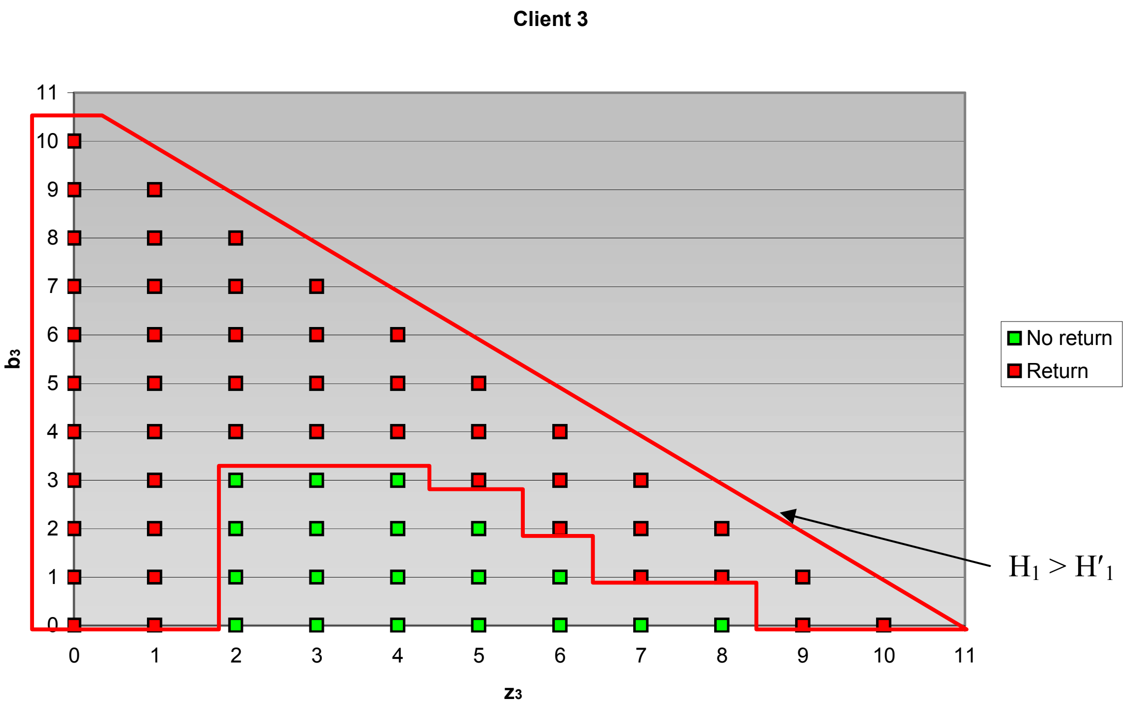


Figure 7.10. The combined threshold graph for pickup and delivery for Client 3.

From this figure it is clear that the function $f_3(z,b)$ is not non-increasing, at least with respect to z_3. The shape of the (z_3,b_3) area for which the vehicle should continue to the next client is reasonable. Let's consider the case for which $b_3 = 2$; then for $z_3 = 0$ *or* 1 the vehicle does not carry enough product to supply the next customer, although it has adequate space to carry returned items. Due to the limited stock on board the vehicle should return to the depot to reload. As the stock rises $z_3 \in \{2,\ldots,5\}$ there is no need to refill; in addition there is space left to carry returned items. Thus there is no need for a depot return. For values of $z_3 \geq 6$ there is not enough space for returned items, and thus the vehicle should return to the depot to unload. Let's now consider the case for which $z_3 = 4$. For low values of $b_3 \in \{0,\ldots,3\}$ both the product stock and the space left on the vehicle are adequate to serve the next customer(s) and no depot return is necessary. However, for values of $b_3 \geq 4$ there is not adequate space left, and thus the vehicle should return to the depot to unload.

The performance of the algorithm was also found to be within acceptable levels. For a test-problem of 10 customer points, 1 delivery product, 1 pickup item, and total vehicle capacity of 10 units, the algorithm derived the minimum expected cost, as well as the individual combined threshold values per customer point, within 338 seconds (=5,6 min). The experiments were run on a PC equipped with Intel Pentium IV, at 2.4 GHz, and 512 MB of RAM.

In order to further assess the performance of the algorithm, a large number of other problem test cases were generated and run. Three different problem test cases are plotted in Figure 7.11. Each case represents 10,000 randomly generated problems in total (1000 problems per client). Each point represents the number on clients in the particular example, ranging from 5 to 15 clients. The first test case concerns a vehicle with a total capacity (for both the pickup and delivery products) Q = 5 and is shown in green color. The second test case concerns a vehicle with a total capacity Q = 10 and is shown in blue color. The third test case concerns a vehicle with a total capacity Q = 15 and is shown in red color. The demand distributions for each problem were generated randomly (with *z* mean values close to 50% of the vehicle capacity and *b* mean values close to 30% of the vehicle capacity).

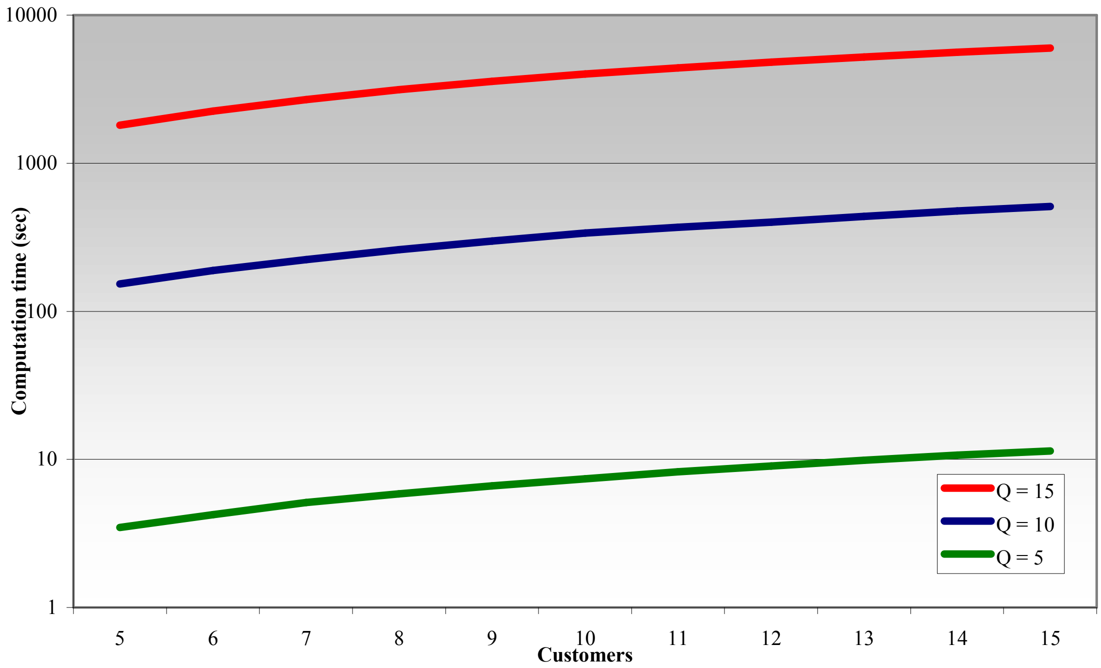


Figure 7.11. Performance results of the algorithm for the pickup and delivery case.

From Figure 7.11 it can be clearly seen that for a given number of customers, the increase of the vehicle capacity resulted in an almost exponential increase in the computation time of the algorithm (note that the scale of the y-axis of the graph is on a logarithmic scale). On the other hand, if the capacity of the vehicle is kept constant, and the number of customers is increased, the computation time also increases significantly. In order to analyze this trend further, ten different test cases were run with Q = 10, for five to fifty customer points and the results are shown in Figure 7.12. Each point in this figure represents the average time for the 1000 problems. The figure indicates that the increase in the computation time is linear with the number of customers.

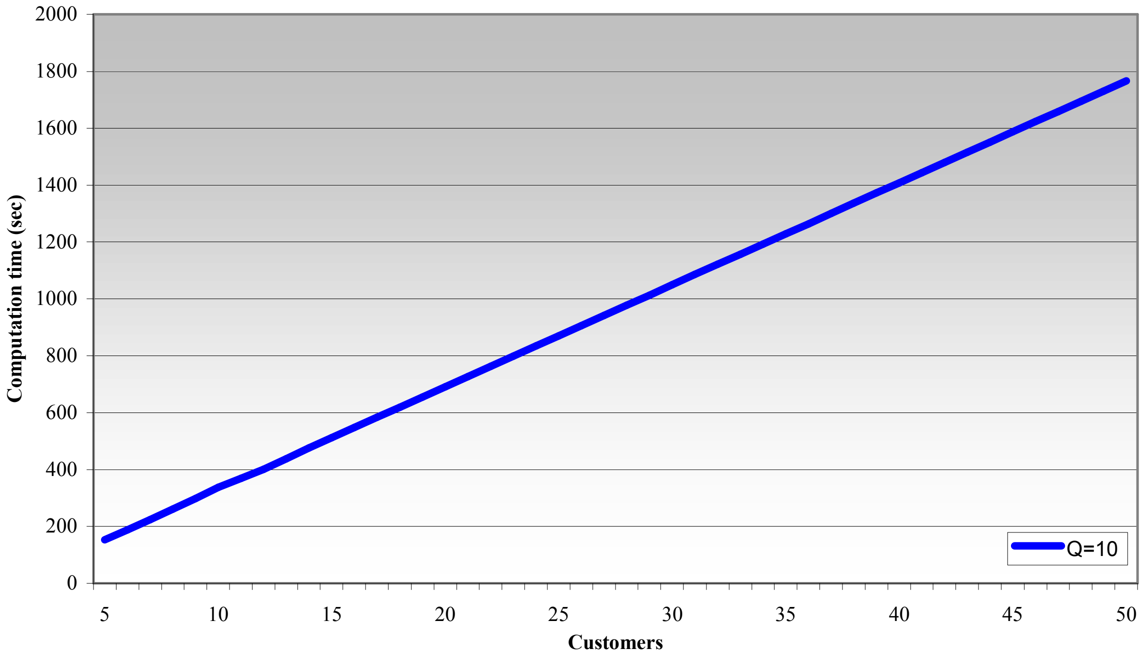


Figure 7.12. Performance results with up to 50 customer points.

For the 50-customer instance the algorithm took approximately 1766 seconds (= 29 min) in order to obtain the solution of the problem.

7.5 Conclusions

In this chapter we presented the Pickup and Delivery case of the Stochastic Vehicle Routing with Depot Returns for Stock Replenishment Problem (SVRDRP). In this case the vehicle not only delivers products to the customers but it also picks up returned items from each customer (e.g. damaged goods, or empty packaging). The objective is to serve all customers by minimizing travel cost under random customer demand.

The characteristics of the problem were presented, together with a new method to determine the minimum expected cost, as well as the optimal decision after serving each customer. In this case there are additional issues to be considered. First, upon visiting the depot, an additional decision needs to be made regarding the quantity of stock to be loaded onto the vehicle. This is due to the fact that it may not prove cost effective to load the vehicle to its full capacity, as there needs to be some space available in order to be able to accommodate the returned items as well without causing unnecessary depot returns. If the depot return comes after serving client $j+1$ then the quantity of product loaded onto the vehicle can be such that the demand of client $j+1$ is fully satisfied (either for delivery product or for returned items). However, if the depot return comes before visiting client $j+1$ then another (subsequent) return to the depot will be necessary in case the product quantity loaded onto the vehicle (or space left for the items to be picked up) is not adequate to fully satisfy the client's demand. During the second return, the decision to load a specific quantity of the product to be delivered is an informed one, since the demand of the client $j+1$ is fully known.

In addition, computational results have shown that a theorem analogous to Theorems 1 and 2 of Chapter 6 does not hold for the pickup and delivery case of the SVRDRP. This is due to the different characteristics of this case, as there is a direct relation between the stock levels of the delivery product and the space available for the pickup items (as the first declines the second rises).

By solving a large number of randomly generated problems, it was found that the increase of the vehicle capacity results in an almost exponential increase in the computational time of the algorithm. On the other hand, if the capacity of the vehicle is kept constant, and the number of customers is increased, the computation time increases linearly by the number of customers.

Chapter 8

Conclusions and Future Research Directions

8.1 Conclusions

In this work we presented, modeled, solved, and analyzed several important cases of the single Vehicle Routing with Depot Returns Problem (VRDRP). We also highlighted the practical importance of this problem in the Ex-van sales and other cases, including material handling system routing.

The objective of the Vehicle Routing with Depot Returns Problem (VRDRP) is to minimize cost (distance) while serving all customers in a predefined sequence with a single vehicle. The analysis of the problem showed that its complexity increases exponentially with the number of customers. In addition, a dynamic programming algorithm (DPA) inspired by the work of Yang *et al.* (2000) and Manfrin *et al.* (2004) was developed to solve the problem to optimality in efficient computational times. The VRDRP formed the foundation of this dissertation and was gradually enhanced in the following chapters as shown in Figure 8.1 below.

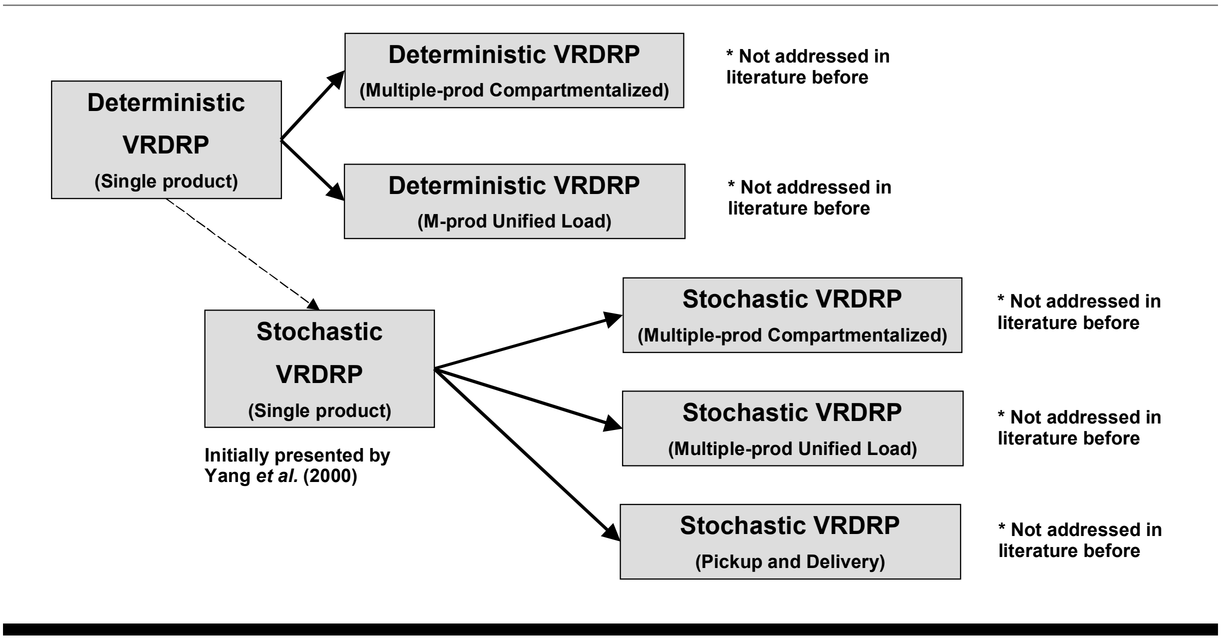


Figure 8.1. The enhancements of the VRDRP.

We enhanced the deterministic VRDRP by studying two significant variations of the problem: (i) The case of multiple-product deliveries in which each product is stored in its own compartment in the vehicle, and (ii) the case of multiple-product deliveries in which all products are stored together in the vehicle's single compartment. The mathematical models, as well as new efficient algorithms that solve these problems to optimality were developed and presented. For the case of compartmentalized load (Problem 1) 3000 problems were created and solved. The number of customers in these problems ranged from 5 to 50, while the number of products was equal to 2, 3 and 4. For the case of unified load (Problem 2) we generated 2000 problems following the procedure described above. In this case the number of customers ranged from 5 to 1000, while the number of products ranged from 2 to 5. Based on the experimental results of the multiple-product problem, it has been demonstrated that the complexity of the compartmentalized case is significantly higher from that of the unified load case. Furthermore, the computational time for the compartmentalized case increases with a rate less than exponential with respect to the number of customers, while for the unified load case it increases linearly. In addition, for the compartmentalized case, the computational time increases exponentially with respect to the number of product types. This is due to the increase of the number of combinations to be examined for each additional product.

In the Stochastic enhancement of the Vehicle Routing with Depot Returns Problem (SVRDRP), the customer demands have been assumed to be independent random variables with known distributions. The SVRDRP has been initially presented and solved by Yang *et al.* (2000). We analyzed this problem to further determine the effect of the variance of the demand on the minimum expected cost function. It was found that the expected cost of the route increases almost linearly with the standard deviation of the demand. Thus, in the Ex-van business case, the consistency of Sales affects the distribution costs directly. Secondly, the problem was analyzed in order to determine the interaction between the mean and the variance of the demand. It was found that this interaction is significant, and, thus, in practice the randomness affects the expected cost more for vehicles with lower capacity.

We studied the multiple products extension of the SVRDRP. Again, we focused on two cases; The compartmentalized and the unified load case. For both cases we presented the characteristics of each problem, novel methods to determine the minimum expected cost, and the theoretical results that permit one to determine the optimal decision after serving each customer. Both cases were addressed using dynamic programming, and for both it was proven that there exists an appropriate threshold function for each customer that distinguishes two regions in the space of possible loads (after serving the customer): The region of loads for which the optimal decision is to return to the depot, and the region for which the optimal decision is to continue to the next customer. For both cases, through the execution of a large number of randomly generated problems it was concluded that the increase of the vehicle capacity results in an almost exponential increase in the computational time of the algorithm. On the other hand, if the capacity of the vehicle is kept constant, and the number of customers is increased, the computational time increases linearly with the number of customers. The unified load case proved to be significantly more complex than the compartmentalized load one, as expected. This is mainly due to the increased complexity of the formulation, which includes additional minimization steps in order to identify the replenishment stock for each customer site.

Finally, we investigated the Pickup and Delivery case of the SVRDRP. In this case the vehicle not only delivers products to customers but it also picks up returned items from each customer (e.g. damaged goods, or empty packaging). In this case there are additional issues to be considered. First, upon visiting the depot, an additional decision needs to be made regarding the quantity of stock to be loaded onto the vehicle. This is due to the fact that it may not be cost effective to load the vehicle to its full capacity, since there needs to be some space available to accommodate the returned items as well without causing unnecessary depot returns. Secondly, it was found that due to the direct relationship between the stock of the delivery product and the space available for the pickup items, the properties of this case are quite different from those of the multiple product one. The characteristics of the problem were presented, together with a novel method to determine the minimum expected cost. By solving a large number of randomly generated problems, it was found that in this case again the increase of the vehicle capacity results in an almost exponential increase in the computational time of the algorithm. On the other hand, if the capacity of the vehicle is kept constant and the number of customers is increased, the computation time increases linearly with the number of customers.

This work has produced a decision support framework, which can be utilized in fixed routing operations (including Ex-van sales and material handling systems within a manufacturing plant) in an urban setting environment (it is not practical to use the above framework in intercity environments since the significantly larger arcs among the cities make the depot returns unfavourable). The routing approach developed in this work can be implemented within a Fleet Management System (as shown in Figure 8.2), which performs both initial routing (planning) and dynamic adjustments of the initial plan responding to disturbances of the environment (execution). The planning part of the system may use the proposed algorithms, based on the characteristics of the distribution environment, in order to develop initial optimal routes. These plans may be distributed to the vehicle drivers either manually (paper-based operation) or electronically (via on-board telematic equipment and communications through GSM/GPRS networks). In case of unexpected events (such as traffic congestion, unexpected delays, etc.) during the plan execution, suitable variations of the proposed algorithms may be employed for dynamic re-planning.

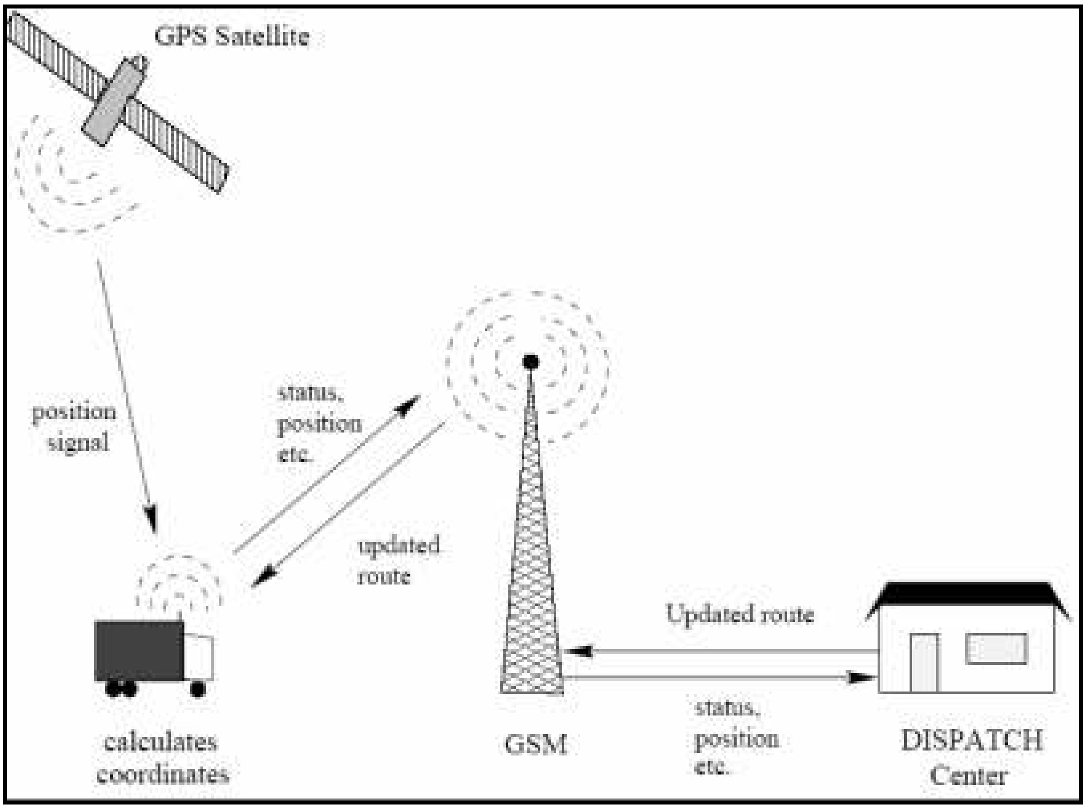


Figure 8.2 A typical Fleet Management System Architecture (Larsen, 1999).

Thus, based on the algorithms described in this work, the operation of a wide variety of cases (deterministic or stochastic demand, single or multiple products, delivery or pickup & delivery) can be improved significantly: Ad-hoc non-optimal decisions are eliminated, minimizing total operating expenses, and increasing the overall productivity of the distribution fleet.

8.2 Future Research Issues

In this work we addressed a single vehicle operation. Typically the Ex-van business model refers to a fleet of vehicles (*n* vehicles), therefore bearing similarities to the Vehicle Routing Problem (VRP). The multiple vehicle case can be transformed to the single vehicle case by assigning *a priori* a cluster of clients to each vehicle (cluster-first-route-second policy). In practice, these assignments are typically made based on historical data and field experience, or simply based on the geographical locations of the customer sites. An interesting problem would be to integrate the customer assignment and the stochastic VRDRP problems is a unified model that would provide globally optimal, or near-optimal, solutions. This would resemble more a route-first-cluster-second policy. The comparison of the two policies, the cluster-first-route-second vs. the route-first-cluster-second, also posses significant issues. The customer network as well as the demand distribution characteristics, could affect the solutions derived by each policy, and it would therefore be challenging to identify which policy would be more efficient over a large amount of randomly generated problem instances.

A second enhancement of this work would be to consider customer time-windows. For example, it is common in the Ex-van business model for each customer (or at least for the large customers) to restrict delivery within pre-agreed time-windows. These windows are usually strict (also called hard time-windows) and if an Ex-van vehicle misses the time-window, the vehicle will not be allowed to serve the respective customer, missing a potentially valuable sales opportunity. The predefined sequence of our problem is compatible with the time-window characteristic, since spatial and temporal sequencing may be related. Further investigation of the hard time-window case is interesting and may have significant practical implications.

Service levels also present an opportunity for further research work. It is common in the Ex-van business model to have different customers, some with a high profile (higher valued customers, with frequent, and/or high value orders) and some with low (with infrequent, low value orders). These customers should be treated differently, in a practical scenario. In its realization, each customer may be allocated a *service factor S* that will be taken into account in the objective function. Depending on the value of *S*, one could give priority to customers with a high *S* over others with a low *S*, or even skip visiting some low-*S* customers in favor of more high profile ones. Modeling of this case may be inspired by the so-called *Orienteering Problem*. Providing optimal, or near-optimal, solution methods will be both interesting and of significant practical value.

Chapter 9

References

9.1 Introduction

This Chapter is dedicated to all the people mentioned in the reference list. Without their work, ideas, vision and ultimately indirect help, the completion of this doctoral would have been impossible. We truly hope that this work will also become the inspiration to take research in this particularly interesting field, another step further.

9.2 Reference List

- Alfredo F., Montane T., Galvao R.D. (2006), "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service", Computers & Operations Research 33 (3), pp.595-619.
- Ashour, S., Vega, J. F and Parker, R.G. (1972), "A heuristic for Traveling Salesman Problems", Transportation Research 6, pp. 187-195.
- Assad A.A. (1988). Modeling and implementation issues, in: Vehicle Routing: Methods and Studies.B.L. Golden and A.A. Assad, editors. North-Holland, Amsterdam.
- Augerat P., Belenguer J. M., Benavent E., Corberán A., Naddef D. (1998), "Separating capacity constraints in the CVRP using tabu search", European Journal of Operational Research, 106 (2-3), pp.546-557.
- Alba E. and Dorronsoro B. (2006), "Computing nine new best-so-far solutions for Capacitated VRP with a cellular Genetic Algorithm", Information Processing Letters 98 (6), pp.225-230.
- Ballou R. H. (1999), Business Logistics Management. 4th International Edition, Prentice-Hall International Inc., New Jersey.
- Barbarosoglu, G., Ozgur, D. (1999), "A tabu search algorithm for the vehicle routing problem", Computers and Operations Research 26, pp. 255-270.
- Beasley, J.E. and Christofides, N., (1997), "Vehicle routing with a sparse feasibility graph", European Journal of Operational Research 98, pp. 499-511.
- Bersimas, D.J., (1988), " Probabilistic combinatorial optimization problems", Ph.D. Thesis, Report No. 193, Operations Research Center, Massachusetts Institute of Technology, Cambridge, M.A.
- Bertsimas D.J., (1992), "A vehicle routing problem with stochastic demand", Operations Research 40 (3), pp. 574-585.

- Bertsimas D. J, Jaillet P, and Odoni A, (1990), “A priori optimization”, Operations Research 38, pp.1019–1033.
- Birattari M., C. Blum, M. Sampels, O. Rossi-Doria, M. Chiarandini, L. Pauqete, T.Schiavinotto, L. Bianchi, M. Mastrolilli, T. Bousonville, H. Juillie, and D. Huber (2005), “Hybrid metaheuristics for the vehicle routing problem with stochastic demands” Technical Report, IDSIA/USI-SUPSI, Dalle Molle Institute for Artificial Intelligence.
- Bodin, L., Golden B., Assad A., Ball M. (1983), "Routing and Scheduling of Vehicles and Crews - The State of the Art", Computers and Operations Research 10 (2), pp. 63-212.
- Bompadre A., Dror M., Orlin J.B. (2006), “Improved bounds for vehicle routing solutions”, Discrete Optimization 3 (4), pp.299-316.
- Bowers, M., Noon, C.E., Thomas, B. (1996), “A parallel implementation of the TSSP +1 decomposition for the capacity-constrained vehicle routing problem”, Computers Operations Research 23, no.7, pp. 723-732.
- Breedam, V. (1995), “Improvement heuristics for the vehicle routing problem based on simulated annealing”, European Journal of Operational Research 86, pp. 480–490.
- Carraway, R.L., Morin, T.L. and Moskowitz, H. (1989), “Generalized dynamic programming for stochastic combinatorial optimization”, Operations Research 37, pp. 819-829.
- Casco D.O., B.L. Golden and E.A. Wasil (1988). Vehicle routing with backhauls: Models, algorithms and case studies, in: Vehicle Routing: Methods and Studies. B.L. Golden and A.A. Assad editors. North-Holland, Amsterdam.
- Christofides, N., Eilon, S. (1969), “An algorithm for the vehicle dispatching problem”, Operational Research Quarterly 20, pp. 309–318.

- Christofides N. (1979). The traveling salesman problem, Combinatorial Optimization. Edited by N. Christofides, R. Mingozzi, P. Toth and C. Sandi. Wiley, New York.
- Cornuejols G. and Nemhauser GL. (1978). ‘‘Tight bounds for Christofides traveling salesman heuristic’’. Mathematical Programming, 14 (1), pp.116-121.
- Coy, S. P., Golden, B. L., Wasil, E. A. (1999), “A computational study of smoothing heuristics for the traveling salesman problem”, European Journal of Operational Research, 124, pp. 15-27.
- Clarke G. and J. Wright. (1964), “Scheduling of vehicles from a central depot to a number of delivery points”. Operations Research 12 (4), pp.568-581.
- Dantzig G.B. and Ramser, J.H. (1959), “The truck dispatching problem”, Management Science 6 (1), pp.80-91.
- Dror, M., Trudeau. P., (1986), “Stochastic vehicle routing with modified savings algorithm”, European Journal of Operational Research 23, pp. 228–235.
- Dror, M., Ball, M., (1987), “Inventory/Routing: Reduction from an Annual to a Short-Period Problem”, Naval Research Logistics 34, pp. 891-905.
- Dror, M., Laporte, G., Trudeau, P. (1989), “Vehicle Routing with Stochastic Demands: Properties and Solution Frameworks”, Transportation Science 23 (3), pp.166-176.
- Dror, M., Trudeau, P. (1990), “Split delivery routing”, Naval Research Logistics 37 (3), pp. 383–402.
- Dror, M. (1993), “Modeling vehicle routing with uncertain demands as a stochastic program: Properties of the corresponding solution”, European Journal of Operational Research 64 (3), pp. 432-441.
- Dror, M. Laporte, G. Trudeau, P. (1994) “Vehicle routing with split deliveries”, Discrete Applied Mathematics 50, pp. 239-254.

- Dror, M., Trudeau, P. (1986), "Stochastic vehicle routing with modified savings algorithm", European Journal of Operational Research 23 (2), pp.228-235.
- Fagerholt, K. (2001), "Ship scheduling with soft time windows: An optimization based approach", European Journal of Operational Research 131, pp. 559-571.
- Fisher, M.L. Jaikumar, R. (1981), "A Generalized Assignment Heuristic for Vehicle Routing", Networks 11, pp.109-124.
- Fisher, M.L. (1994), "Optimal Solution of Vehicle routing Problems using minimum k-trees", Operations Research 42, No.4, pp. 626-642.
- Garey, M. R., Johnson, D. S. (1979), "Computers and intractability: a guide to the theory of NP completeness", New York: W. H. Freeman & Co
- Gendreau M., Laporte G., Seguin, R. (1995), "An exact algorithm for the vehicle routing problem with stochastic demands and customers", Transportation Science 29 (2), pp.143-155.
- Gendreau M., Laporte G., Seguin, R. (1996), "Stochastic Vehicle Routing", European Journal of Operational Research 88, pp.3-12.
- Gendreau M., Laporte G., Hertz, A. (1997), "An Approximation Algorithm for the Traveling Salesman Problem with Backhauls", *Operations Research* 45 (4), pp.639-641.
- Giaglis, G. M., Minis, I., Tatarakis, A., Zeimpekis, V. (2004) "Minimizing Logistics Risk through Real-Time Vehicle Routing and Mobile Technologies: Research To-Date and Future Trends", International Journal of Physical Distribution and Logistics Management, Vol. 34 (9), pp.749-764.
- Gillett, B.E., Miller, L. R. (1974), "A heuristic for the vehicle-dispatch problem", Operations Research 22, pp. 340-349.
- Glover, F., Gutin, G., Yeo, A., Zverovich, A. (2001), "Construction heuristics for the asymmetric TSP", European Journal of Operational Research 129, pp. 555-568

- Golden B.L., Wasil E.A., (1987), “Computerized vehicle routing in the soft drink industry”, Operations Research 35 (1), pp.6-17.
- Golden, B. L., Assad, A. A. (Eds.) (1988), "Vehicle Routing: Methods and Studies, Studies in Management Science and Systems", Vol. 16, Elsevier Science Publications, Amsterdam, pp. 7-45.
- Hachicha, M., Hodgson, M.J., Laporte, G., Semet, F. (2000), “Heuristics for the multi-vehicle covering problem”, Computers and Operations Research 27, pp. 29–42.
- Ho S. C., Haugland D. (2004), “A tabu search heuristic for the vehicle routing problem with time windows and split deliveries”, Computers & Operations Research 31 (12), pp.1947-1964.
- Ioannou, G., Kritikos, M., Prastacos, G. (2003), “A problem generator-solver heuristic for vehicle routing with soft time windows”, Omega 31, pp. 41-53.
- Jaillet, P. (1987), “Stochastic routing problems”, in: G. Andreatta, F. Mason and P. Serafini (eds.), Stochastics in Combinatorial Optimization, World Scientific, New Jersey.
- Jaillet, P. and Odoni, A.R. (1988), “The probabilistic vehicle routing problem”, in B.L. Golden and A.A. Assad (eds.), Vehicle routing: Models and Studies, North-Holland, Amsterdam.
- Jezequel, A. (1985), “Probabilistic Vehicle routing Problems”, M.Sc. Dissertation, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, M.A.
- Kall P. (1982), “Stochastic Programing”, European Journal of Operational Research 10, pp.125-130.
- Koksalan, M., Sural, H., Kirca, O. (1995), “A location-distribution application for a beer company”, European Journal of Operational Research 80, pp. 16-24.
- Kyriakidis, E.G., Dimitrakos, T.D. (2007), “Single vehicle routing problem with a predefined customer sequence and stochastic continuous demands, The Mathematical Scientist, *In Press*
- Laporte G., Louveaux F., Mercure H. (1989), “Models and Exact solutions for a class of stochastic location-routing problems”, European Journal of Operational Research 39, pp.71-78.

- Laporte G., Louveaux F. (1990), “Formulations and bounds for the stochastic capacitated vehicle routing problem with uncertain supplies”, in J.J. Gabszewicz, J-F. Richard and L.A. Wolsey (eds.), Economic Decision-making: Games, Econometrics, and Optimisation, North-Holland, Amsterdam.
- Laporte, G. (1992), “The vehicle routing problem: An overview of exact and approximate algorithms”, European Journal of Operational Research 59 (3), pp. 345–358.
- Laporte G. and Osman I.H. (1995), “Routing problems: A bibliography”, Annals of Operation Research 61 (1), 227-262.
- Laporte G. and Osman I.H. (1996), “Metaheuristics: A bibliography”, Annals of Operation Research 63 (5), 513-623.
- Laporte, G. Gengreau, M., Potvin, J., Semet, F., (2000), “Classical and modern heuristics for the vehicle routing problem”, International Transactions in Operational Research, pp. 285–300.
- Larsen, J. (1999), “Vehicle Routing with Time-Windows – Finding optimal solutions efficiently”, Technical paper, In press.
- Lenstra, J.K, Rinnooy Kan AHG. (1981), “Complexity of vehicle routing and scheduling problems”, Networks, 11 (2), pp. 221–227.
- Longo H., Poggi de Aragão M., Uchoa E. (2006), “Solving capacitated arc routing problems using a transformation to the CVRP”, Computers & Operations Research 33 (6), pp.1823-1837.
- Manfrin M., Birattari M., Blum C., Sampels M., Rossi-Doria O., Chiarandini M., Pauqete L., Schiavinotto T., Bianchi L., Mastrolilli M., Bousonville T., Juillie H., and Huber D. (2004), “Report for task 7: Experimental comparison”. Internal document of the Metaheuristics Network.
- Manfrin M., Birattari M., Blum C., Sampels M., Rossi-Doria O., Chiarandini M., Pauqete L., Schiavinotto T., Bianchi L., and Mastrolilli M., (2004), “Report for task 8: Final analysis”. Internal document of the Metaheuristics Network.

- Min, H., Jayaraman, V., Srivastava, R. (1998), “Combined location-routing problems: A synthesis and future research directions”, European Journal of Operational Research 108, pp. 1-15.
- Modesti, P., Sciomachen, A. (1998), “A utility measure for finding multiobjective shortest paths in urban multimodal transportation networks”, European Journal of Operational Research 111, pp. 495-508.
- Nagy G., Salhi S. (2005), “Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries”, European Journal of Operational Research 162 (1), pp. 126-141.
- Nanry, W. P. and Barnes, J. W. (2000), “Solving the pickup and delivery problem with time windows using reactive tabu search”, Transportation Research Part B 34, pp. 107-121.
- Or, I. (1976), “Travelling salesman-type combinatorial problems and their relation to the logistics of blood banking”, PhD thesis, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston IL.
- Pisinger D. and Ropke S. (2007), “A general heuristic for vehicle routing problems”, Computers & Operations Research 34 (8), pp. 2403-2435.
- Rego, C., Rucairol, C. (1995), “Using tabu search for solving a dynamic multi-terminal truck dispatching problem”, European Journal of Operational Research 83, pp. 411-429.
- Reimann, M., Doerner, K., Hartl, R.F. (2003), D-Ants: Savings Based Ants divide and conquer the vehicle routing problem, Computers & Operations Research, Vol. 31 (4), pp. 563-591.
- Renaud, J., Boctor, F.F., Ouenniche, J. (2000), “A heuristic for the pickup and delivery traveling salesman problem”, Computers & Operations Research 27, pp. 905-916.
- Righini, G. (2000) "Approximation algorithms for the vehicle routing problem with pick-up and delivery", Note del Polo - Ricerca 33, Polo Didattico e di Ricerca di Crema, Università degli Studi di Milano.

- Roberts D. and Hadjiconstantinou E. (1990), "A computational approach to the vehicle routing problem with stochastic demands", In P. Borne, M. Ksouri, and A. El Kamel, editors, Computational Engineering in Systems Applications, pp.139–144.
- Robuste, F., Daganzo, C.F., Souleyrette II, R.R. (1990), "Implementing vehicle routing models", Transportation Research part B 24B (4), pp. 263-286.
- Savelsbergh, M.W.P. (1995), "The General Pickup and Delivery Problem", Transportation Science 29, pp. 17-29.
- Secomandi N. (1998). Exact and Heuristic Dynamic Programming Algorithms for the Vehicle Routing Problem with Stochastic Demands. PhD thesis, University of Houston.
- Secomandi, N. (2000), "Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands", Computers & Operations Research 27, pp. 1201-1225.
- Secomandi, N. (2001), "A rollout policy for the vehicle routing problem with stochastic demands, Operations Research 49 (5), pp. 796-802.
- Smith, D. K.(1991), "Dynamic Programming: A Practical Introduction", Ellis Horwood, Chichester.
- Stewart W.R. and Golden B. L. (1983), "Stochastic Vehicle Routing: A comprehensive approach", European Journal of Operation Research 14, pp.371-385.
- Tarantilis, C.D., Diakoulaki, D., Kiranoudis, C.T. (2004), "Combination of geographical information system and efficient routing algorithms for real life distribution operations", European Journal of Operational Research, Volume 152 (2), pp. 437-453.
- Tarantilis, C.D., Kiranoudis, C.T. (2002), "Using a spatial decision support system for solving the vehicle routing problem", Information & Management 39, pp. 359-375.
- Tarantilis C.D., Ioannou G., Prastacos G. (2005), "Advanced vehicle routing algorithms for complex operations management problems", Journal of Food Engineering 70 (3), pp.455-471.

- Tatarakis A., I. Minis, E Kyriakides (2005). "The Vehicle Routing with Depot Returns for Stock Replenishment Problem (VRDRSRP)". Working Paper
- Toth P. and Vigo D. (2002), "Models, relaxations and exact approaches for the capacitated vehicle routing problem", Discrete Applied Mathematics 123 (1-3), pp.487-512.
- Toth, P., Vigo, D. (2002a), "Models, relaxations and exact approaches for the capacitated vehicle routing problem", Discrete Applied Mathematics 123, pp. 487-512.
- Toth, P., Vigo, D. (2002b), "The vehicle routing problem", (Ed.), Philadelphia; Siam, 2002.
- Tsiribas P. (2006), "The extensions of Vehicle Routing with Depot Returns for Stock Replenishment Problem", University of the Aegean, Greece.
- Trudeau P. and Dror M. (1992), "Stochastic Inventory Routing: Route Design with Stockouts and Route Failures", Transportation Science 26 (3), pp.171-184.
- Van der Poort, E.S., Libura, M., Sierksma, G., Van den Veen, J.A.A. (1999), "Solving the k-best traveling salesman problem", Computers & Operations Research 26, pp. 409-425.
- Vliet, A.V., Boender, C.G.E., Rinnooy Kan, A.H.G. (1992), "Interactive Optimization of Bulk Sugar Deliveries", Interfaces 22, pp. 4-14.
- Yang W-H., Mathur K., Ballou R.H. (2000), "Stochastic Vehicle Routing Problem with Restocking", Transportation Science 34, pp.99-112.
- Yellow, P. (1970), "A computational modification to the savings method of vehicle scheduling", Operational Research Quarterly 21, pp. 281–283.
- Waters, C.D.J., (1989), "Vehicle-scheduling problems with uncertainty and omitted customers", Journal of the Operational Research Society 40, pp.1099-1108.
- Wren, A., Holliday, A. (1972), "Computer scheduling of vehicles from one or more depots to a number of delivery points", Operational Research Quarterly 23, pp. 333–344.

Appendices

Appendix A

A.1 The Yang *et al* optimal solution

According to Yang *et al.* (2000), in order to identify an optimal route approach for a single vehicle, it is first necessary to develop an efficient procedure to evaluate a particular route, that is, to find its expected cost under an optimal restocking policy. Upon service completion at customer *j*, suppose the vehicle has a remaining load *z*, and let $f_j(z)$ denote the total expected cost from node *j* onward. If S_j represents the set of all possible loads that a vehicle can have after service completion at customer *j*, then, $f_j(z)$ for $z \in S_j$ satisfies the dynamic programming recursion:

$$f_j(z) = \text{Minimum} \begin{cases} c_{j,j+1} + \sum_{k:\xi^k \le z} f_{j+1}(z-\xi^k)p_{j+1,k} + \sum_{k:\xi^k > z} [b + 2c_{j+1,0} + f_{j+1}(z+Q-\xi^k)]p_{j+1,k} \\ \\ c_{j,0} + c_{0,j+1} + \sum_{k=1}^{m} f_{j+1}(Q-\xi^k)p_{j+1,k} \end{cases} \quad (A.1)$$

with the boundary condition:

$$f_n(z) = c_{n0} \qquad z \in S_n \quad (A.2)$$

In Eq. (A.1) the upper term in the minimization represents the expected cost of going directly to the next customer, whereas the lower term represents the expected cost of the restocking action. Dynamic programming is used to recursively determine the optimal policy. The properties of the optimal policy are derived as follows.

LEMMA 1.

$$f_j(z) \le f_j(Q) + 2c_{0j} \qquad \text{for all} \qquad z \in S_j$$

Proof. From Eq. (A.1),

$$f_j(z) \le c_{j,0} + c_{0,j+1} + \sum_{k=1}^{m} f_{j+1}(Q-\xi^k)p_{j+1,k} \quad (A.3)$$

Also because C satisfies the triangular inequality, Eq. (A.1) gives

$$f_j(Q) \le c_{j,j+1} + \sum_{k=1}^{m} f_{j+1}(Q-\xi^k)\, p_{j+1,k} \tag{A.4}$$

Combining Eq. (A.3) and (A.4) results in

$$\begin{aligned} f_j(z) &\le c_{j,0} + c_{0,j+1} - c_{j,j+1} + f_j(Q) \\ &\le c_{j,0} + c_{0,j} + c_{j,j+1} - c_{j,j+1} + f_j(Q) \\ &\qquad \text{(by the triangular inequality)} \\ &= 2c_{0,j} + f_j(Q) \end{aligned}$$

This seemingly simple lemma constitutes a key element to the proof of the following theorem.

THEOREM 1. *For each customer j, there exists a quantity h_j, such that the optimal decision, after serving node j, is to continue to node j + 1 if $z \ge h_j$, or return to the depot if $z < h_j$.*

Proof. To prove this theorem, we first show by induction that for all $z \in S_j$, $f_j(z)$ is a non-increasing function. That is, for $z_1, z_2 \in S_j$ and $z_1 < z_2$,

$$f_j(z_1) \ge f_j(z_2)$$

At terminal stage n, $f_n(z) = c_{n0}$ is independent of z. Hence $f_n(z)$ is monotonically non-increasing with respect to $z \in S_n$. We will now prove that, if $f_{j+1}(z)$ is monotonically non-increasing with respect to $z \in S_{j+1}$, then $f_j(z)$ is also monotonically non-increasing with respect to $z \in S_j$.

Let $H_j(z)$ and $H'_j(z)$ denote the values of the upper and lower terms inside the minimisation in Eq. (A.1). Then for $z_1, z_2 \in S_j$ and $z_1 < z_2 \le Q$, $H_j(z_1) - H_j(z_2)$, after some simplification, can be written as

$$\begin{aligned} H_j(z_1) - H_j(z_2) = &\sum_{k:\xi^k \le z_1} [f_{j+1}(z_1-\xi^k) - f_{j+1}(z_2-\xi^k)]\, p_{j+1,k} + \sum_{k:z_1<\xi^k \le z_2} [b + 2c_{j+1,0} + f_{j+1}(z_1+Q-\xi^k) - f_{j+1}(z_2-\xi^k)]\, p_{j+1,k} + \\ &+ \sum_{k:\xi^k \ge z_2} [f_{j+1}(z_1+Q-\xi^k) - f_{j+1}(z_2+Q-\xi^k)]\, p_{j+1,k} \end{aligned}$$

Because $f_{j+1}(z)$ is monotonically non-increasing, the first and the third summation in the above equation is positive. Hence,

$$H_j(z_1) - H_j(z_2) \geq \sum_{k:z_1<\xi^k \leq z_2} [b + 2c_{j+1,0} + f_{j+1}(z_1 + Q - \xi^k) - f_{j+1}(z_2 - \xi^k)] p_{j+1,k}$$

Using LEMMA 1 and the monotonicity of $f_{j+1}(z)$, it is now easy to show that $H_j(z_1) - H_j(z_2) \geq 0$. Hence $H_j(z)$ is a monotonically non-increasing function and $f_j(z)$ is the minimum of a non-increasing function $H_j(z)$ and a constant function $H'_j(z)$, hence it is monotonically non-increasing with respect to $z \in S_j$. Moreover there exists h_j, such that the optimal decision, after serving node j, is to continue to node $j+1$ if $z \geq h_j$, or return to the depot if $z < h_j$. Note that $h_j = 0$ if $H_j(z) \leq H'_j(z)$ for all $z \in S_j$, and $h_j = Q$ if $H'_j(z) \leq H_j(z)$) for all $z \in S_j$.

The main implication of Theorem 1 is that, practically, it is easy to implement because it provides a simple policy for the driver to follow. Second, this result can be used in the efficient algorithmic implementation of the dynamic programming recursion. In particular, at each stage of the dynamic programming, the algorithm computes $H'_j(z)$. Then it computes $H_j(z)$ in descending order of z until it exceeds the value of $H'_j(z)$. The last value of z for which $H_j(z) \leq H'_j(z)$ is the threshold h_j for this customer. If z is higher than the threshold the vehicle can proceed to the next customer site, otherwise it will return to the depot.

Appendix B

B.1 The Multiple product compartmentalized case formulation – 3 products

We assume that the vehicle is divided into 3 sections and each section is suitable for one type of product only. Using the notation of Section 6.2.1, the dynamic programming formulation of the 3-product problem is given below:

$$f_j(z_{1j}, z_{2j}, z_{3j}) = \min$$

$$\begin{cases}
\begin{aligned}
& c_{j,j+1} + \\
& + \sum_{k_1:\xi_1^{k_1} \le z_{1j}} \sum_{k_2:\xi_2^{k_2} \le z_{2j}} \sum_{k_3:\xi_3^{k_3} \le z_{3j}} f_{j+1}(z_{1j} - \xi_1^{k_1}, z_{2j} - \xi_2^{k_2}, z_{3j} - \xi_3^{k_3}) p_{j+1,k_1} p_{j+1,k_2} p_{j+1,k_3} + \\
& + \sum_{k_1:z_{1j} < \xi_1^{k_1}} \sum_{k_2:\xi_2^{k_2} \le z_{2j}} \sum_{k_3:\xi_3^{k_3} \le z_{3j}} \left[2c_{j+1,0} + f_{j+1}(Q_1 - \xi_1^{k_1} + z_{1j}, Q_2, Q_3)\right] p_{j+1,k_1} p_{j+1,k_2} p_{j+1,k_3} + \\
& + \sum_{k_1:\xi_1^{k_1} \le z_{1j}} \sum_{k_2:z_{2j} < \xi_2^{k_2}} \sum_{k_3:\xi_3^{k_3} \le z_{3j}} \left[2c_{j+1,0} + f_{j+1}(Q_1, Q_2 - \xi_2^{k_2} + z_{2j}, Q_3)\right] p_{j+1,k_1} p_{j+1,k_2} p_{j+1,k_3} + \\
& + \sum_{k_1:\xi_1^{k_1} \le z_{1j}} \sum_{k_2:\xi_2^{k_2} \le z_{2j}} \sum_{k_3:z_{3j} < \xi_3^{k_3}} \left[2c_{j+1,0} + f_{j+1}(Q_1, Q_2, Q_3 - \xi_3^{k_3} + z_{3j})\right] p_{j+1,k_1} p_{j+1,k_2} p_{j+1,k_3} + \\
& + \sum_{k_1:z_{1j} < \xi_1^{k_1}} \sum_{k_2:z_{2j} < \xi_2^{k_2}} \sum_{k_3:\xi_3^{k_3} \le z_{3j}} \left[2c_{j+1,0} + f_{j+1}(Q_1 - \xi_1^{k_1} + z_{1j}, Q_2 - \xi_2^{k_2} + z_{2j}, Q_3)\right] p_{j+1,k_1} p_{j+1,k_2} p_{j+1,k_3} + \\
& + \sum_{k_1:z_{1j} < \xi_1^{k_1}} \sum_{k_2:\xi_2^{k_2} \le z_{2j}} \sum_{k_3:z_{3j} < \xi_3^{k_3}} \left[2c_{j+1,0} + f_{j+1}(Q_1 - \xi_1^{k_1} + z_{1j}, Q_2, Q_3 - \xi_3^{k_3} + z_{3j})\right] p_{j+1,k_1} p_{j+1,k_2} p_{j+1,k_3} + \\
& + \sum_{k_1:\xi_1^{k_1} \le z_{1j}} \sum_{k_2:z_{2j} < \xi_2^{k_2}} \sum_{k_3:z_{3j} < \xi_3^{k_3}} \left[2c_{j+1,0} + f_{j+1}(Q_1, Q_2 - \xi_2^{k_2} + z_{2j}, Q_3 - \xi_3^{k_3} + z_{3j})\right] p_{j+1,k_1} p_{j+1,k_2} p_{j+1,k_3} + \\
& + \sum_{k_1:z_{1j} < \xi_1^{k_1}} \sum_{k_2:z_{2j} < \xi_2^{k_2}} \sum_{k_3:z_{3j} < \xi_3^{k_3}} \left[2c_{j+1,0} + f_{j+1}(Q_1 - \xi_1^{k_1} + z_{1j}, Q_2 - \xi_2^{k_2} + z_{2j}, Q_3 - \xi_3^{k_3} + z_{3j})\right] p_{j+1,k_1} p_{j+1,k_2} p_{j+1,k_3} \\
\\
& c_{j,0} + c_{0,j+1} + \\
& \sum_{k_1=1}^{m_1} \sum_{k_2=1}^{m_2} \sum_{k_3=1}^{m_3} f_{j+1}(Q_1 - \xi_1^{k_1}, Q_2 - \xi_2^{k_2}, Q_3 - \xi_3^{k_3}) p_{j+1,k_1} p_{j+1,k_2} p_{j+1,k_3}
\end{aligned}
\end{cases}$$

As it can be clearly observed, the first part of the minimization equation consists of eight distinct terms (the eight rows after the $c_{j,j+1}$ row).

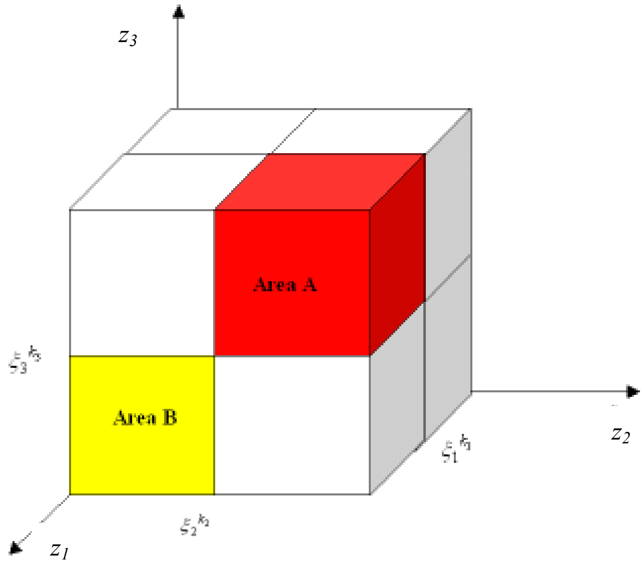


Figure B.1. The solution space per customer point for three products.

The first of these terms represents the cost that is incurred if the vehicle proceeds to the next customer directly and the stock of all three items on board is sufficient to fully satisfy the demand of the next customer (Area A shown in red in Figure B.1). The next three terms represent the cases in which the stock of one product (z_1, z_2, or z_3) is not sufficient to satisfy the demand of the next customer, while the stock of the other two products is.

The next three terms represent the cases in which the stock of only one product (z_1, z_2, or z_3) is sufficient to satisfy the demand of the next customer, while the stock of the other two products is not. The eighth term represents the cost that is incurred if the vehicle proceeds to the next customer directly and the stock on board of all products is not sufficient to fully satisfy the demand of the next customer.

The second part of the equation represents the cost that is incurred if the vehicle does not go to the next customer directly, but rather first visits the Depot, refills its stock of all products, and then visits the next customer in the sequence.

Appendix C

C.1 Demand Distribution for the Multiple Product Delivery: Compartmentalized Load

For product 1

Customer \ Demand	0	1	2	3	4	5
1	0,01	0,7	0,1	0,1	0,08	0,01
2	0,01	0,6	0,2	0,1	0,08	0,01
3	0,01	0,7	0,2	0,05	0,03	0,01
4	0,01	0,5	0,2	0,2	0,08	0,01
5	0,01	0,2	0,6	0,1	0,08	0,01

For product 2

Customer \ Demand	0	1	2	3	4	5
1	0,01	0,1	0,1	0,7	0,08	0,01
2	0,01	0,2	0,6	0,1	0,08	0,01
3	0,01	0,03	0,05	0,2	0,7	0,01
4	0,01	0,8	0,1	0,05	0,03	0,01
5	0,01	0,08	0,1	0,2	0,6	0,01

Vehicle Capacity = (5,5)

Appendix D

D.1 Demand Distribution for the Multiple Product Delivery: Unified Load

Demand for Product 1	Demand for Product 2	Customer 1	Customer 2	Customer 3	Customer 4	Customer 5
10	0	0,001	0,001	0,001	0,001	0,001
9	1	0,001	0,001	0,001	0,001	0,001
9	0	0,001	0,001	0,001	0,001	0,001
8	2	0,001	0,001	0,001	0,001	0,001
8	1	0,001	0,001	0,001	0,001	0,001
8	0	0,001	0,001	0,001	0,001	0,001
7	3	0,001	0,001	0,001	0,001	0,001
7	2	0,001	0,001	0,001	0,001	0,001
7	1	0,001	0,001	0,001	0,001	0,001
7	0	0,001	0,001	0,001	0,001	0,001
6	4	0,001	0,001	0,001	0,001	0,001
6	3	0,001	0,001	0,001	0,001	0,001
6	2	0,001	0,001	0,001	0,001	0,001
6	1	0,001	0,001	0,001	0,001	0,001
6	0	0,001	0,001	0,001	0,001	0,001
5	5	0,001	0,001	0,001	0,001	0,001
5	4	0,001	0,001	0,001	0,001	0,001
5	3	0,001	0,001	0,001	0,001	0,001
5	2	0,001	0,001	0,001	0,001	0,001
5	1	0,001	0,001	0,001	0,001	0,001
5	0	0,001	0,001	0,001	0,001	0,001
4	6	0,001	0,001	0,001	0,001	0,001
4	5	0,001	0,001	0,001	0,001	0,001
4	4	0,001	0,001	0,001	0,001	0,001
4	3	0,001	0,001	0,001	0,001	0,001
4	2	0,001	0,001	0,001	0,001	0,001
4	1	0,001	0,001	0,001	0,001	0,001
4	0	0,001	0,001	0,001	0,001	0,001
3	7	0,001	0,001	0,001	0,001	0,001
3	6	0,001	0,001	0,001	0,001	0,001
3	5	0,001	0,001	0,001	0,001	0,001
3	4	0,001	0,001	0,001	0,001	0,001
3	3	0,001	0,001	0,001	0,001	0,001
3	2	0,001	0,001	0,001	0,001	0,001
3	1	0,001	0,001	0,001	0,1	0,001
3	0	0,001	0,001	0,001	0,001	0,001
2	8	0,001	0,001	0,001	0,001	0,001
2	7	0,001	0,001	0,001	0,001	0,001
2	6	0,001	0,001	0,001	0,001	0,001
2	5	0,001	0,001	0,001	0,001	0,001

2	4	0,001	0,001	0,001	0,001	0,6
2	3	0,001	0,001	0,2	0,001	0,001
2	2	0,2	0,1	0,001	0,2	0,01
2	1	0,01	0,21	0,01	0,01	0,01
2	0	0,001	0,001	0,001	0,001	0,001
1	9	0,001	0,001	0,001	0,001	0,001
1	8	0,001	0,001	0,001	0,001	0,001
1	7	0,001	0,001	0,001	0,001	0,001
1	6	0,001	0,001	0,001	0,001	0,001
1	5	0,001	0,001	0,001	0,001	0,001
1	4	0,001	0,001	0,7	0,002	0,001
1	3	0,7	0,01	0,01	0,01	0,2
1	2	0,01	0,6	0,01	0,01	0,101
1	1	0,01	0,01	0,01	0,6	0,01
1	0	0,01	0,01	0,001	0,01	0,01
0	10	0,001	0,001	0,001	0,001	0,001
0	9	0,001	0,001	0,001	0,001	0,001
0	8	0,001	0,001	0,001	0,001	0,001
0	7	0,001	0,001	0,001	0,001	0,001
0	6	0,001	0,001	0,001	0,001	0,001
0	5	0,001	0,001	0,001	0,001	0,001
0	4	0,001	0,001	0,001	0,001	0,001
0	3	0,001	0,001	0,001	0,001	0,001
0	2	0,001	0,001	0,001	0,001	0,001
0	1	0,001	0,001	0,001	0,001	0,001
0	0	0,001	0,001	0,001	0,001	0,001

Vehicle Capacity = 10

Appendix E

E.1 Demand Distribution for the Pickup and Delivery example

For product to be delivered

Customer \ Demand	0	1	2	3	4	5
1	0,01	0,1	0,1	0,7	0,08	0,01
2	0,01	0,2	0,6	0,1	0,08	0,01
3	0,01	0,03	0,05	0,2	0,7	0,01
4	0,01	0,8	0,1	0,05	0,03	0,01
5	0,01	0,08	0,1	0,2	0,6	0,01

For product to be picked up

Customer \ Demand	0	1	2	3	4	5
1	0,01	0,7	0,1	0,1	0,08	0,01
2	0,01	0,6	0,2	0,1	0,08	0,01
3	0,01	0,7	0,2	0,05	0,03	0,01
4	0,01	0,5	0,2	0,2	0,08	0,01
5	0,01	0,2	0,6	0,1	0,08	0,01

Vehicle Capacity = 5

Appendix F

F.1 The Characteristics of Equations 6.10 and 7.1

In this Appendix we will show that Equations 6.10 and 7.1 define a proper dynamic programming recursion. In particular we will show that the way these equations have been written is a compact form of equivalent but more extended dynamic programming equations. This will be done by focusing on Equation 6.10. The discussion is completely analogous for Equation 7.1. Taking part (a) of Equation 6.10 we have:

$$\left\{\begin{array}{l} c_{j,j+1} + \sum\limits_{k_1:\xi^{k_1}\le z_1} \sum\limits_{k_2:\xi^{k_2}\le z_2} f_{j+1}(z_1-\xi^{k_1}, z_2-\xi^{k_2}) p_{j+1}(k_1,k_2) + \\ + \sum\limits_{k_1:z_1<\xi^{k_1}} \sum\limits_{k_2:\xi^{k_2}\le z_2} \left[2c_{j+1,0} + \min\limits_{\xi^{k_1}-z_1\le\theta_1\le Q} f_{j+1}(\theta_1-(\xi^{k_1}-z_1), Q-\theta_1)\right] p_{j+1}(k_1,k_2) + \\ + \sum\limits_{k_1:\xi^{k_1}\le z_1} \sum\limits_{k_2:z_2<\xi^{k_2}} \left[2c_{j+1,0} + \min\limits_{\xi^{k_2}-z_2\le\theta_2\le Q} f_{j+1}(\theta_2, Q-\theta_2-(\xi^{k_2}-z_2))\right] p_{j+1}(k_1,k_2) + \\ + \sum\limits_{k_1:z_1<\xi^{k_1}} \sum\limits_{k_2:z_2<\xi^{k_2}} \left[2c_{j+1,0} + \min\limits_{\xi^{k_1}-z_1\le\theta_3\le Q-(\xi^{k_2}-z_2)} f_{j+1}(\theta_3-(\xi^{k_1}-z_1), Q-\theta_3-(\xi^{k_2}-z_2))\right] p_{j+1}(k_1,k_2) \end{array}\right\} (1) \Leftrightarrow$$

$$\Leftrightarrow \left\{\begin{array}{l} c_{j,j+1} + \sum\limits_{k_1:\xi^{k_1}\le z_1} \sum\limits_{k_2:\xi^{k_2}\le z_2} f_{j+1}(z_1-\xi^{k_1}, z_2-\xi^{k_2}) p_{j+1}(k_1,k_2) + \\ + \min\limits_{\xi^{k_1}-z_1\le\theta_1\le Q} \sum\limits_{k_1:z_1<\xi^{k_1}} \sum\limits_{k_2:\xi^{k_2}\le z_2} \left[2c_{j+1,0} + f_{j+1}(\theta_1-(\xi^{k_1}-z_1), Q-\theta_1)\right] p_{j+1}(k_1,k_2) + \\ + \min\limits_{\xi^{k_2}-z_2\le\theta_2\le Q} \sum\limits_{k_1:\xi^{k_1}\le z_1} \sum\limits_{k_2:z_2<\xi^{k_2}} \left[2c_{j+1,0} + f_{j+1}(\theta_2, Q-\theta_2-(\xi^{k_2}-z_2))\right] p_{j+1}(k_1,k_2) + \\ + \min\limits_{\xi^{k_1}-z_1\le\theta_3\le Q-(\xi^{k_2}-z_2)} \sum\limits_{k_1:z_1<\xi^{k_1}} \sum\limits_{k_2:z_2<\xi^{k_2}} \left[2c_{j+1,0} + f_{j+1}(\theta_3-(\xi^{k_1}-z_1), Q-\theta_3-(\xi^{k_2}-z_2))\right] p_{j+1}(k_1,k_2) \end{array}\right\} (2) \Leftrightarrow$$

$$\Leftrightarrow \min_{\substack{\xi^{k_1}-z_1\le\theta_1\le Q \\ \xi^{k_2}-z_2\le\theta_2\le Q \\ \xi^{k_1}-z_1\le\theta_3\le Q-(\xi^{k_2}-z_2)}} \left\{\begin{array}{l} c_{j,j+1} + \sum\limits_{k_1:\xi^{k_1}\le z_1} \sum\limits_{k_2:\xi^{k_2}\le z_2} f_{j+1}(z_1-\xi^{k_1}, z_2-\xi^{k_2}) p_{j+1}(k_1,k_2) + \\ + \sum\limits_{k_1:z_1<\xi^{k_1}} \sum\limits_{k_2:\xi^{k_2}\le z_2} \left[2c_{j+1,0} + f_{j+1}(\theta_1-(\xi^{k_1}-z_1), Q-\theta_1)\right] p_{j+1}(k_1,k_2) + \\ + \sum\limits_{k_1:\xi^{k_1}\le z_1} \sum\limits_{k_2:z_2<\xi^{k_2}} \left[2c_{j+1,0} + f_{j+1}(\theta_2, Q-\theta_2-(\xi^{k_2}-z_2))\right] p_{j+1}(k_1,k_2) + \\ + \sum\limits_{k_1:z_1<\xi^{k_1}} \sum\limits_{k_2:z_2<\xi^{k_2}} \left[2c_{j+1,0} + f_{j+1}(\theta_3-(\xi^{k_1}-z_1), Q-\theta_3-(\xi^{k_2}-z_2))\right] p_{j+1}(k_1,k_2) \end{array}\right\} (3)$$

The equivalence of sums (1) and (2) is true since all terms in sum (1) are linear with positive known coefficients, which are the joint probability mass functions $p_{j+1}(k_1,k_2)$. Thus the minimum of each linear term in sum (2) is equal to the corresponding term of sum (1), in which the minimum is internal to the term.

The equivalence of sums (2) and (3) is true since each sum is linear and each term within sum (2) is positive. Thus the global minimum of sum (3) is equal to the addition of the local minima in sum (2).

Similarly, for the second part of Equation (6.10) we have:

$$
\left\{\begin{array}{l}
c_{j,0}+c_{0,j+1}+ \\
\min\limits_{0\le\theta_4\le Q}\left\{\begin{array}{l}
\sum\limits_{\xi^{k_1}\le\theta_4}\ \sum\limits_{\xi^{k_2}\le Q-\theta_4} f_{j+1}(\theta_4-\xi^{k_1},Q-\theta_4-\xi^{k_2})\,p_{j+1}(k_1,k_2)+ \\
+\sum\limits_{\theta_4<\xi^{k_1}}\ \sum\limits_{\xi^{k_2}\le Q-\theta_4}\left[2c_{j+1,0}+\min\limits_{\xi^{k_1}-\theta\le s_1\le Q} f_{j+1}(s_1-(\xi^{k_1}-\theta_4),Q-s_1)\right]p_{j+1}(k_1,k_2)+ \\
+\sum\limits_{\xi^{k_1}\le\theta_4}\ \sum\limits_{Q-\theta_4<\xi^{k_2}}\left[2c_{j+1,0}+\min\limits_{0\le s_2\le 2Q-\theta_4-\xi^{k_2}} f_{j+1}\{s_2,Q-s_2-[\xi^{k_2}-(Q-\theta_4)]\}\right]p_{j+1}(k_1,k_2)
\end{array}\right\}
\end{array}\right\}(4)\Leftrightarrow
$$

$$
\Leftrightarrow\left\{\begin{array}{l}
c_{j,0}+c_{0,j+1}+ \\
\min\limits_{0\le\theta_4\le Q}\left\{\begin{array}{l}
\sum\limits_{\xi^{k_1}\le\theta_4}\ \sum\limits_{\xi^{k_2}\le Q-\theta_4} f_{j+1}(\theta_4-\xi^{k_1},Q-\theta_4-\xi^{k_2})\,p_{j+1}(k_1,k_2)+ \\
+\min\limits_{\xi^{k_1}-\theta\le s_1\le Q}\sum\limits_{\theta_4<\xi^{k_1}}\ \sum\limits_{\xi^{k_2}\le Q-\theta_4}\left[2c_{j+1,0}+f_{j+1}(s_1-(\xi^{k_1}-\theta_4),Q-s_1)\right]p_{j+1}(k_1,k_2)+ \\
+\min\limits_{0\le s_2\le 2Q-\theta_4-\xi^{k_2}}\sum\limits_{\xi^{k_1}\le\theta_4}\ \sum\limits_{Q-\theta_4<\xi^{k_2}}\left[2c_{j+1,0}+f_{j+1}\{s_2,Q-s_2-[\xi^{k_2}-(Q-\theta_4)]\}\right]p_{j+1}(k_1,k_2)
\end{array}\right\}
\end{array}\right\}(5)\Leftrightarrow
$$

$$
\Leftrightarrow\min\limits_{\substack{0\le\theta_4\le Q\\ \xi^{k_1}-\theta\le s_1\le Q\\ 0\le s_2\le 2Q-\theta_4-\xi^{k_2}}}\left\{\begin{array}{l}
c_{j,0}+c_{0,j+1}+ \\
\left\{\begin{array}{l}
\sum\limits_{\xi^{k_1}\le\theta_4}\ \sum\limits_{\xi^{k_2}\le Q-\theta_4} f_{j+1}(\theta_4-\xi^{k_1},Q-\theta_4-\xi^{k_2})\,p_{j+1}(k_1,k_2)+ \\
+\sum\limits_{\theta_4<\xi^{k_1}}\ \sum\limits_{\xi^{k_2}\le Q-\theta_4}\left[2c_{j+1,0}+f_{j+1}(s_1-(\xi^{k_1}-\theta_4),Q-s_1)\right]p_{j+1}(k_1,k_2)+ \\
+\sum\limits_{\xi^{k_1}\le\theta_4}\ \sum\limits_{Q-\theta_4<\xi^{k_2}}\left[2c_{j+1,0}+f_{j+1}\{s_2,Q-s_2-[\xi^{k_2}-(Q-\theta_4)]\}\right]p_{j+1}(k_1,k_2)
\end{array}\right\}
\end{array}\right\}(6)
$$

The equivalence of sums (4) and (5) is true since all terms in sum (4) are linear with positive known coefficients, which are the joint probability mass functions $p_{j+1}(k_1,k_2)$. Thus the minimum of each linear term in sum (5) is equal to the corresponding term of sum (4), in which the minimum is internal to the term.

The equivalence of sums (5) and (6) is true since each sum is linear and each term within sum (5) is positive. Thus the global minimum of sum (6) is equal to the addition of the local minima in sum (5).

Bringing part (a) and (b) of Equation 6.10 back together using sums (3) and (6) we obtain:

$$
f_j(z_1,z_2)=\min
\left\{
\begin{array}{l}
\displaystyle\min_{\substack{\xi^{k_1}-z_1\le\theta_1\le Q\\ \xi^{k_2}-z_2\le\theta_2\le Q\\ \xi^{k_1}-z_1\le\theta_3\le Q-(\xi^{k_2}-z_2)}}
\left\{
\begin{array}{l}
c_{j,j+1}+\displaystyle\sum_{k_1:\xi^{k_1}\le z_1}\ \sum_{k_2:\xi^{k_2}\le z_2} f_{j+1}(z_1-\xi^{k_1},z_2-\xi^{k_2})p_{j+1}(k_1,k_2)+\\
+\displaystyle\sum_{k_1:z_1<\xi^{k_1}}\ \sum_{k_2:\xi^{k_2}\le z_2}\left[2c_{j+1,0}+f_{j+1}(\theta_1-(\xi^{k_1}-z_1),Q-\theta_1)\right]p_{j+1}(k_1,k_2)+\\
+\displaystyle\sum_{k_1:\xi^{k_1}\le z_1}\ \sum_{k_2:z_2<\xi^{k_2}}\left[2c_{j+1,0}+f_{j+1}(\theta_2,Q-\theta_2-(\xi^{k_2}-z_2))\right]p_{j+1}(k_1,k_2)+\\
+\displaystyle\sum_{k_1:z_1<\xi^{k_1}}\ \sum_{k_2:z_2<\xi^{k_2}}\left[2c_{j+1,0}+f_{j+1}(\theta_3-(\xi^{k_1}-z_1),Q-\theta_3-(\xi^{k_2}-z_2))\right]p_{j+1}(k_1,k_2)
\end{array}
\right\}\\
\displaystyle\min_{\substack{0\le\theta_4\le Q\\ \xi^{k_1}-\theta\le s_1\le Q\\ 0\le s_2\le 2Q-\theta_4-\xi^{k_2}}}
\left\{
\begin{array}{l}
c_{j,0}+c_{0,j+1}+\\
\left(
\begin{array}{l}
\displaystyle\sum_{\xi^{k_1}\le\theta_4}\ \sum_{\xi^{k_2}\le Q-\theta_4} f_{j+1}(\theta_4-\xi^{k_1},Q-\theta_4-\xi^{k_2})p_{j+1}(k_1,k_2)+\\
+\displaystyle\sum_{\theta_4<\xi^{k_1}}\ \sum_{\xi^{k_2}\le Q-\theta_4}\left[2c_{j+1,0}+f_{j+1}(s_1-(\xi^{k_1}-\theta_4),Q-s_1)\right]p_{j+1}(k_1,k_2)+\\
+\displaystyle\sum_{\xi^{k_1}\le\theta_4}\ \sum_{Q-\theta_4<\xi^{k_2}}\left[2c_{j+1,0}+f_{j+1}\{s_2,Q-s_2-[\xi^{k_2}-(Q-\theta_4)]\}\right]p_{j+1}(k_1,k_2)
\end{array}
\right)
\end{array}
\right\}
\end{array}
\right\}\Leftrightarrow
$$

$$
\Leftrightarrow \min_{\substack{\xi^{k_1}-z_1\le\theta_1\le Q \\ \xi^{k_2}-z_2\le\theta_2\le Q \\ \xi^{k_1}-z_1\le\theta_3\le Q-(\xi^{k_2}-z_2) \\ 0\le\theta_4\le Q \\ \xi^{k_1}-\theta\le s_1\le Q \\ 0\le s_2\le 2Q-\theta_4-\xi^{k_2}}}
\left\{
\begin{array}{l}
c_{j,j+1}+\displaystyle\sum_{k_1:\xi^{k_1}\le z_1}\ \sum_{k_2:\xi^{k_2}\le z_2} f_{j+1}(z_1-\xi^{k_1},z_2-\xi^{k_2})\,p_{j+1}(k_1,k_2)+ \\
+\displaystyle\sum_{k_1:z_1<\xi^{k_1}}\ \sum_{k_2:\xi^{k_2}\le z_2}\Big[2c_{j+1,0}+f_{j+1}(\theta_1-(\xi^{k_1}-z_1),Q-\theta_1)\Big]p_{j+1}(k_1,k_2)+ \\
+\displaystyle\sum_{k_1:\xi^{k_1}\le z_1}\ \sum_{k_2:z_2<\xi^{k_2}}\Big[2c_{j+1,0}+f_{j+1}(\theta_2,Q-\theta_2-(\xi^{k_2}-z_2))\Big]p_{j+1}(k_1,k_2)+ \\
+\displaystyle\sum_{k_1:z_1<\xi^{k_1}}\ \sum_{k_2:z_2<\xi^{k_2}}\Big[2c_{j+1,0}+f_{j+1}(\theta_3-(\xi^{k_1}-z_1),Q-\theta_3-(\xi^{k_2}-z_2))\Big]p_{j+1}(k_1,k_2) \\
\\
\qquad\qquad\qquad\qquad c_{j,0}+c_{0,j+1}+ \\
\displaystyle\sum_{\xi^{k_1}\le\theta_4}\ \sum_{\xi^{k_2}\le Q-\theta_4} f_{j+1}(\theta_4-\xi^{k_1},Q-\theta_4-\xi^{k_2})\,p_{j+1}(k_1,k_2)+ \\
+\displaystyle\sum_{\theta_4<\xi^{k_1}}\ \sum_{\xi^{k_2}\le Q-\theta_4}\Big[2c_{j+1,0}+f_{j+1}(s_1-(\xi^{k_1}-\theta_4),Q-s_1)\Big]p_{j+1}(k_1,k_2)+ \\
+\displaystyle\sum_{\xi^{k_1}\le\theta_4}\ \sum_{Q-\theta_4<\xi^{k_2}}\Big[2c_{j+1,0}+f_{j+1}\{s_2,Q-s_2-[\xi^{k_2}-(Q-\theta_4)]\}\Big]p_{j+1}(k_1,k_2)
\end{array}
\right\}
$$

The latter equation is a proper dynamic programming recursion.