ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΝΑΥΠΗΓΩΝ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΠΕΡΙΟΧΗ ΘΑΛΑΣΣΙΩΝ ΜΕΤΑΦΟΡΩΝ

ΜΑΘΗΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΜΕ ΚΕΡΔΗ

Διπλωματική Εργασία

ΤΟΥΡΛΟΥΠΗΣ ΒΑΣΙΛΕΙΟΣ

Νοέμβριος 2007

ΑΘΗΝΑ

Τουρλούπης Βασίλειος

ΜΑΘΗΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΜΕ ΚΕΡΔΗ

Νοέμβριος 2007

Διπλωματική Εργασία

Σχολή Ναυπηγών Μηχανολόγων Μηχανικών

Συγγραφέας: Βασίλειος Τουρλούπης

Επιβλέπων: Δημήτριος Β. Λυρίδης

ΑΘΗΝΑ

ΠΡΟΛΟΓΟΣ

Η παρούσα εργασία αποτελεί τη διπλωματική μου εργασία στα πλαίσια των σπουδών μου στη Σχολή Ναυπηγών Μηχανολόγων Μηχανικών. Αισθάνομαι την ανάγκη να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου Δημήτριο Β. Λυρίδη, του οποίου η ηθική και υλική συμπαράσταση συντέλεσε σημαντικά στην περάτωση της παρούσας εργασίας. Επίσης, θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου, που στάθηκαν δίπλα μου όλο αυτό το διάστημα.

Κατάλογος Σχημάτων

Κατάλογος Διαγραμμάτων

Κατάλογος Πινάκων

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ

Η παρούσα μελέτη ασχολείται με το πρόβλημα δρομολόγησης οχημάτων με κέρδη, που είναι γνωστό στην διεθνή ορολογία ως Vehicle Routing Problem with Profits (VRPP). Πρόκειται για ένα πρόβλημα εύρεσης της διαδρομής που πρέπει να ακολουθήσει ένας στόλος οχημάτων ώστε να εξυπηρετήσει ένα δεδομένο σύνολο πελατών, με στόχο το βέλτιστο κέρδος. Το VRPP είναι μία επέκταση του Vehicle Routing Problem, στο οποίο το ζητούμενο είναι η ελαχιστοποίηση του κόστους.

Το VRP, αλλά και το VRPP, είναι ένα πρόβλημα που αφορά τομείς μεταφοράς, διανομής και logistics. Σε αρκετούς τομείς της αγοράς, το κόστος μεταφοράς κατά ένα μεγάλο ποσοστό του προστίθεται στην τιμή του προϊόντος. Έτσι, η χρήση υπολογιστικών μεθόδων με σκοπό την μείωση του κόστους μεταφοράς μπορεί να σημαίνει μείωση του ολικού κόστους μέχρι και έως 20%. Έτσι, το VRP βρίσκει συχνά εφαρμογή σε εταιρείες διανομών και courier.

Το συγκεκριμένο πρόβλημα (VRP) έχει γίνει αντικείμενο αρκετών μελετών τα τελευταία 40 χρόνια, λόγω της πρακτικής του χρησιμότητας αλλά και της δυσκολίας του. Είναι ένα πρόβλημα ακέραιης βελτιστοποίησης τύπου NP-hard, που σημαίνει ότι ο υπολογιστικός χρόνος που απαιτεί η επίλυσή του αυξάνει εκθετικά όσο αυξάνει το μέγεθος του προβλήματος.

Πρέπει να αναφέρουμε πως κατά την εφαρμογή του VRP στον πραγματικό κόσμο συχνά εμφανίζονται δευτερεύοντες περιορισμοί. Το αποτέλεσμα είναι η δημιουργία πολυάριθμων υποπροβλημάτων που έχουν ως βάση τους το VRP. Μερικά από τα πιο γνωστά είναι:

- Οι πελάτες πρέπει να ικανοποιηθούν μέσα σε ένα χρονικό διάστημα (VRP with time windows - VRPTW)
- Υπάρχουν περισσότεροι του ενός κεντρικοί κόμβοι (Multiple Depot VRP - MDVRP)
- Οι πελάτες μπορούν να επιστρέψουν μερικά προϊόντα στον κεντρικό κόμβο (VRP with Pick-Up and Delivering - VRPPD)

Το VRPP εισάγει στο VRP τον παράγοντα των εσόδων, δίνοντας νέες διαστάσεις στο πρόβλημα του VRP. Το αντικείμενο βελτιστοποίησης στο VRPP συχνά είναι το κέρδος με αποτέλεσμα πελάτες που δεν συμβάλλουν στην αύξηση του, να αποκλείονται από τις διαδρομές και να μην εξυπηρετούνται. Η παρούσα εργασία παρουσιάζει ένα ολοκληρωμένο τρόπο επίλυσης του VRPP. Εισάγει ένα πρωτότυπο, με βάση όσα γνωρίζουμε, μαθηματικό μοντέλο και μία μέθοδο επίλυσης για 3 είδη του προβλήματος VRPP: Profitable Vehicle Routing Problem, Team Orienteering Problem, Prize-Collecting VRP.

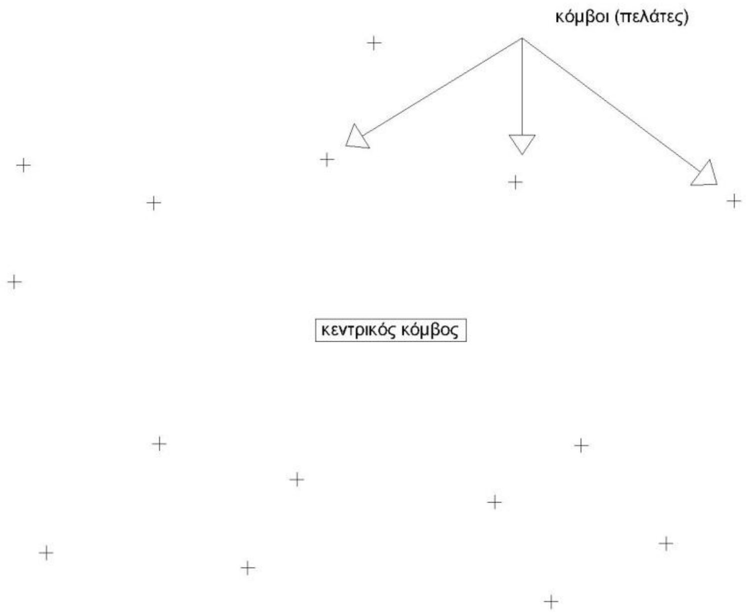
Αυτό που κάνει το VRPP να ξεχωρίζει από το VRP και να το μελετάμε εδώ ως ένα διαφορετικό πρόβλημα είναι, ουσιαστικά, η αβεβαιότητα συμμετοχής των κόμβων στην λύση. Αυτό έχει ως αποτέλεσμα την ύπαρξη διαφορετικού, έως ενός σημείου, μαθηματικού μοντέλου για το πρόβλημα αυτό. Αυτή η αβεβαιότητα συμμετοχής των κόμβων στην λύση έχει και μία άλλη επίπτωση.

Επίσης, εισάγεται ένα καινούργιο πρόβλημα, επέκταση του VRPP, και το αντίστοιχο μαθηματικό μοντέλο. Το ονομάσαμε Fleet-Selective VRPP και πρόκειται για μια παραλλαγή του VRPP στην οποία δεν είναι υποχρεωτικό να συμπεριληφθούν όλα τα οχήματα στην λύση, επιπροσθέτως για την χρησιμοποίηση κάθε οχήματος υπάρχει ένα αρχικό κόστος χρήσης. Το FS-VRPP είναι, δηλαδή, ένας συνδυασμός δύο προβλημάτων. Το πρώτο είναι η δρομολόγηση των οχημάτων και το δεύτερο η επιλογή του βέλτιστου στόλου που θα χρησιμοποιηθεί.

ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ & ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ

Το Πρόβλημα Δρομολόγησης Οχημάτων με Κέρδη

Στο VRPP το ζητούμενο είναι ο καθορισμός των διαδρομών που πρέπει να ακολουθήσει ένας στόλος οχημάτων, με έδρα έναν κεντρικό κόμβο, με σκοπό την εξυπηρέτηση ενός συνόλου γεωγραφικά διασκορπισμένων πελατών, έτσι ώστε το κέρδος να είναι μέγιστο.



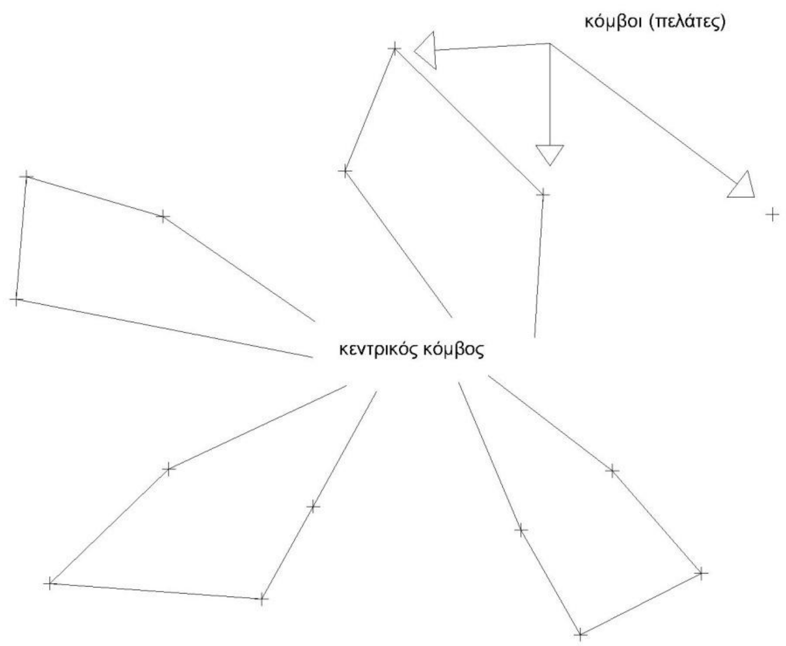
σχήμα 1 Τυπικά δεδομένα για το VRPP

Αναλυτικά, υπάρχει ένας στόλος όμοιων οχημάτων που έχει ως έδρα έναν κεντρικό κόμβο, ο οποίος ουσιαστικά είναι μία πολύ μεγάλη αποθήκη κάποιου προϊόντος. Γύρω από τον κεντρικό κόμβο υπάρχουν διάσκορποι πελάτες (κόμβοι). Οι πελάτες έχουν συγκεκριμένη ζήτηση για το προϊόν του κεντρικού κόμβου. Τα έσοδα από την εξυπηρέτηση κάποιου κόμβου είναι γνωστά και προκαθορισμένα και ανεξάρτητα από την απόσταση του από τον κεντρικό κόμβο.

- Όλα τα οχήματα πρέπει να χρησιμοποιηθούν στην λύση, δεν συμβαίνει το ίδιο και για τους πελάτες.
- Κάθε πελάτης εξυπηρετείται το πολύ μια φορά

- Όλα τα οχήματα ξεκινούν από τον κεντρικό κόμβο και στο τέλος επιστρέφουν σε αυτόν.
- Όλα τα οχήματα έχουν την ίδια χωρητικότητα
- Κάθε διαδρομή μεταξύ κόμβων έχει ένα προκαθορισμένο κόστος, το οποίο είναι ίδιο για το κάθε όχημα
- Ένα όχημα δεν μπορεί να εξυπηρετήσει μόνο ένα ποσοστό της ζήτησης κάποιου πελάτη, στην περίπτωση που τον εξυπηρετεί παραδίδει το 100% της ζήτησής του.

Το ζητούμενο είναι οι διαδρομές που πρέπει να ακολουθήσουν τα οχήματα. Κατά την παρούσα εργασία έγινε μελέτη για 3 είδη του προβλήματος του VRPP. Η κύρια διαφορά τους είναι το αντικείμενο βελτιστοποίησης. Έτσι, στο PVRPP οι διαδρομές είναι αυτές που μεγιστοποιούν το κέρδος, στην περίπτωση του TOP αυτές που μεγιστοποιούν τα έσοδα ενώ στο PCVRPP αυτές που ελαχιστοποιούν το κόστος. Στις περιπτώσεις των TOP και PCVRPP υπάρχουν περιορισμοί ως προς το μέγιστο κόστος ή τις ελάχιστες αποδοχές, αντίστοιχα



σχήμα 2 Τυπικά αποτελέσματα του VRPP

Όπως φαίνεται στο σχήμα 2 στην λύση δεν είναι αναγκαίο να εξυπηρετηθούν όλοι οι κόμβοι Οι κόμβοι που τελικά, που θα εξυπηρετηθούν είναι αυτοί που συμβάλλουν στην αύξηση των κερδών.

Στην πραγματικότητα, τα προβλήματα με τα οποία ασχολούμαστε σε αυτή την εργασία συνδέονται με τα αρχετυπικά, καλά μελετημένα συνδιαστικά προβλήματα του Traveling Salesman Problem (TSP) και VRP. Υπάρχουν πολυάριθμες αναφορές σε αυτά τα προβλήματα, μερικές από τις οποίες έναι των Toth and Vigo (2001) και Gutin and Punnen (2002). Ακολουθούν περιλήψεις δημοσιεύσεων με περιεχόμενο συναφή με την παρούσα εργασία:

Feillet et al (2005): Ασχολείται με το TSP με κέρδη το οποίο είναι μια γενίκευση του TSP, όπου δεν είναι απαραίτητο να εξυπηρετηθούν όλοι οι πελάτες. Ένα κέρδος συνδέεται με κάθε πελάτη. Αντιμετοπίζουν το TSPP ως πρόβλημα δύο διακριτών υποπροβλημάτων βελτιστοποίησης, όπου οι δύο στόχοι είναι η μεγιστοποίηση των αποδοχών και η ταυτόχρονη ελαχιστοποίηση του κόστους. Αναφέρονται, επίσης, στις δύο υποπεριπτώσεις των TOP και Prize-Collecting (PCTSP). Εκτενής αναφορά γίνεται στην υπάρχουσα βιβλιογραφία, σε διάφορες μεθόδους μοντελοποίησης καθώς και στους τρόπους επίλυσης του. Επίσης, δίνεται έμφαση στο ενδιαφέρον που παρουσιάζει αυτή η τάξη προβλημάτων, με σεβασμό σε εφαρμογές και σε θεωρητικά αποτελέσματα. Μετά από προσεκτική ανάγνωση αυτής της εργασίας, κατανοούμε ότι επιλύεται περισσότερο για την περίπτωση ενός οχήματος παρά για την περίπτωση πολλών οχημάτων.

Archetti et al (2005): Ασχολούνται με το TOP ως γενίκευση στην περίπτωση των πολλαπλών διαδρομών του OP. Στις εφαρμογές τους, παρέχεται ένας συνδυασμός πιθανών πελατών και αποκομίζεται κέρδος από την εξυπηρέτηση κάθε πελάτη. Ένας στόλος οχημάτων παρέχεται για την εξυπηρέτηση των πελατών, σε ένα ορισμένο χρονικό όριο. Το κέρδος ενός πελάτη μπορεί να αποκομιστεί από ένα όχημα το πολύ.

Ο σκοπός είναι να εντοπιστούν οι πελάτες που μεγιστοποιούν το συνολικό κέρδος που αποκομίζεται ενώ ικανοποιείται ο ορισμένος χρόνος για κάθε όχημα. Προτείνουν δύο παραλλαγές ενός tabu search και ενός neighborhood search αλγορίθμου βελτιστοποίησης για τη λύση του TOP. Αναφορές γίνονται στην υπολογιστική εμπειρία και στις επιδόσεις του αλγόριθμου.

Tang και Miller-Hooks (2005): Ασχολούνται επίσης με το TOP και το επιλύουν με την ανάπτυξη ενός tabu search heuristic. O tabu search αλγόριθμος ενσωματώνεται σε μία υιοθετητική διαδικασία μνήμης που διαφοροποιείται μεταξύ μικρών και μεγάλων γειτονικών σταδίων κατά τη διάρκεια βελτίωσης της λύσης.

ΜΑΘΗΜΑΤΙΚΗ ΔΙΑΤΥΠΩΣΗ ΚΑΙ ΕΠΙΛΥΣΗ ΤΟΥ ΜΟΝΤΕΛΟΥ

Μαθηματικά μοντέλα του VRPP

Σε αυτό το σημείο πρέπει να γίνει κατανοητό το ότι το μαθηματικό μοντέλο που παρουσιάζουμε προορίζεται για την επίλυση των προβλημάτων που προηγουμένως ορίστηκαν. Πάντως, εύκολα κάποιος με μικρές αλλαγές μπορεί να χρησιμοποιήσει το μοντέλο αυτό για επίλυση παραπλήσιων προβλημάτων (π.χ. για οχήματα διαφορετικών χωρητικοτήτων)

The Profitable Vehicle Routing Problem With Profits

Ονοματολογία

p_i αποδοχές από την εξυπηρέτηση του κόμβου i

b = χωρητικότητα (σε βάρος ή όγκο του κάθε οχήματος).

a_i = το μέγεθος της παράδοσης στον κόμβο i, μετρημένο στην ίδια μονάδα μέτρησης με την χωρητικότητα των οχημάτων

c_{ij} = το κόστος της απευθείας διαδρομής από τον κόμβο i στον κόμβο j (υποθέτουμε $c_{ij} \geq 0$ και $c_{ij}=c_{ji}$ για όλα τα ij)

N ο συνολικός αριθμός των κόμβων

οι μεταβλητές απόφασης είναι:

y_{ik}: 1 αν το όχημα k επισκέπτεται τον κόμβο i, αλλιώς 0.

x_{ijk}: 1 αν το όχημα k κάνει την απευθείας διαδρομή από τον κόμβο i στον κόμβο j, αλλιώς 0.

Ο αντικειμενικός στόχος είναι

$$\max \sum_{ik} p_i y_{ik} - \sum_{ijk} c_{ij} x_{ijk} \text{ , } \forall \text{ i, j, k.} \quad (1)$$

όπου $\sum_{ik} p_i y_{ik}$ οι συνολικές αποδοχές και $\sum_{ijk} c_{ij} x_{ijk}$ το συνολικό κόστος

Περιορισμοί

$$\sum_i x_{ijk} = y_{jk} \quad \forall \text{ j, k.} \quad (2)$$

$$\sum_j x_{ijk} = y_{ik} \quad \forall \text{ i, k.} \quad (3)$$

Οι περιορισμοί (2) & (3) είναι γνωστοί ως περιορισμοί ανάθεσης. Εξασφαλίζουν ‘ότι όσες φορές το όχημα k επισκεφτεί τον κόμβο i τόσες φορές θα πάει καταφθάσει και θα αναχωρήσει από αυτόν

$$u_{ik} - u_{jk} + Nx_{ijk} \leq N - 1, \text{για } i \neq j: i = 2{,}3 \ldots N, j = 2{,}3 \ldots N, k = 1, \ldots K, u_{ik}, u_{jk} \geq 0 \ (4)$$

Ο περιρισμός (4) είναι γνωστός ως περιορισμός εξάλειψης υποδιαδρομών.

$$\sum_i a_i y_{ik} \leq b \quad \forall k \qquad (5)$$

Ο περιορισμός (5) εξασφαλίζει να μην ξεπεραστεί η χωρητικότητα του κάθε οχήματος

$$\sum_{\kappa} y_{ik} \begin{cases} k, & i = 1 \\ binary & i - 2, \ldots, n \end{cases} \qquad (6)$$

Ο περιορισμός (6) εξασφαλίζει την χρησιμοποίηση όλων των οχημάτων στην λύση και ότι κάθε κόμβος (εκτός από τον κεντρικό) θα επισκεφτεί από το πολύ από ένα όχημα.

$$\begin{cases} \mathrm{x_{ijk}} = 0 \text{ ή } 1 \\ \mathrm{y_{ik}} = 0 \text{ ή } 1, \ \text{για } i \neq 1 \end{cases} \qquad (7)$$

i, j = 1, 2, …, N ,όπου N ο αριθμός των κόμβων

k= 1, 2, …, K, όπου K ο αριθμός των οχημάτων

Στο σημείο αυτό πρέπει να γίνει μία αναφορά στον περιορισμό (4). Υποδιαδρομές ονομάζουμε όλες τις διαδρομές που έχουν ως σημείο εκκίνησης οποιοδήποτε κόμβο, εκτός του κεντρικού. Βάση του μαθηματικού μοντέλου υπάρχει περίπτωση ύπαρξη λύσης η οποία περιέχει υποδιαδρομές. Ο περιορισμός (4) έχει ως σκοπό την απόρριψή τους. Ακολουθεί η μαθηματική απόδειξη.

$u_{ik} - u_{jk} + Nx_{ijk} \leq N - 1, \gamma\iota\alpha\ i \neq j: i = 2,3 \dots N, j = 2,3 \dots N, k = 1, \dots K, u_{ik}, u_{jk} \geq 0$ (4)

Ας υποθέσουμε ότι υπάρχουν συνολικά 7 κόμβοι (συμπεριλαμβανομένου και του κεντρικού) και 2 οχήματα. Για να αποδείξουμε ότι όλες οι λύσεις που περιέχουν υποδιαδρομές αποκλείονται από την (4) υποθέτουμε ότι η λύση είναι $x_{151} = x_{211} = x_{431} = x_{341} = x_{521} = x_{162} = x_{612} = 1$. Η διαδρομή του οχήματος 1 συνίσταται από 2 υποδιαδρομές 1-5-2-1 και 3-4-3. Αν πάρουμε την διαδρομή που δεν περιέχει τον κεντρικό κόμβο (1) εφαρμόζοντας την (4) έχουμε:

$$u_3 - u_4 + 7 * x_{341} \leq 7 - 1$$

$$u_4 - u_3 + 7 * x_{431} \leq 7 - 1$$

όπου φυσικά $x_{341}=x_{431}=1$

αθροίζοντας τις δύο ανισότητες παίρνουμε $7(\chi_{341} + \chi_{431}) \leq 12$, κάτι που σαφώς αποκλείει την υποδιαδρομή (όπως και οποιαδήποτε άλλη υποδιαδρομή).

Για να δείξουμε ότι για όλες τις λύσεις που δεν περιέχουν υποδιαδρομές υπάρχουν u_j ικανοποιούν την (4) υποθέτουμε ότι το όχημα 1 κάνει την διαδρομή 1-5-2-1 και το όχημα 2 την διαδρομή 1-3-4-6-1 αυτό σημαίνει πως η λύση είναι $x_{151} = x_{211} = x_{521} = x_{132} = x_{342} = x_{462} = x_{612} = 1$

Ας εξετάσουμε τι συμβαίνει για το όχημα 2. Θέτουμε u_{ik} ίσο με την θέση της διαδρομής στην οποία ο κόμβος επισκέπτεται από το όχημα 2. Η διαδρομή είναι 1-3-4-5-1 έτσι $u_{12}=1$, $u_{32}=2$,$u_{42}=3$,… Οι κόμβοι που δεν επισκέπτονται από το όχημα 2 παίρνουν με τυχαία σειρά τους ακέραιους αριθμούς που υπολείπονται ως το N. Αν πάρουμε ένα τυχαίο x_{ij2} που ισούται με την μονάδα θα έχουμε

$$u_{i2} - u_{j2} + Nx_{ij2} \leq N - 1$$

επειδή το όχημα 2 επισκέπτεται τον κόμβο j αμέσως μετά τον κόμβο k ($x_{ij2}=1$)

$$u_{i2} - u_{j2} = -1$$

έχουμε δηλαδή

$$-1 + N \leq N - 1$$

που ισχύει

Αν τώρα πάρουμε ένα τυχαίο x_{ik2} το οποίο ισούται με μηδέν θα έχουμε

$$u_{i2} - u_{j2} + Nx_{ij2} \leq N - 1$$

δηλαδή $u_{i2}-u_{j2}\leq N-1$ το οποίο ισχύει επειδή $u_{i2}\leq N$ και $u_{j2}>1$ οπότε η μεγαλύτερη τιμή που μπορεί να πάρει το $u_{i2}-u_{j2}$ είναι το N-2 και η ανισότητα ικανοποιείται. Στο ίδιο αποτέλεσμα θα φτάσουμε αν εξετάσουμε και την διαδρομή του οχήματος 1.

The Team Orienteering Problem

Ονοματολογία

p_i αποδοχές από την εξυπηρέτηση του κόμβου i

b = χωρητικότητα (σε βάρος ή όγκο του κάθε οχήματος).

a_i = το μέγεθος της παράδοσης στον κόμβο i, μετρημένο στην ίδια μονάδα μέτρησης με την χωρητικότητα των οχημάτων

c_{ij} = το κόστος της απευθείας διαδρομής από τον κόμβο i στον κόμβο j (υποθέτουμε $c_{ij} \geq 0$ και $c_{ij}=c_{ji}$ για όλα τα ij)

N ο συνολικός αριθμός των κόμβων

οι μεταβλητές απόφασης είναι:

y_{ik}: 1 αν το όχημα k επισκέπτεται τον κόμβο i, αλλιώς 0.

x_{ijk}: 1 αν το όχημα k κάνει την απευθείας διαδρομή από τον κόμβο i στον κόμβο j, αλλιώς 0.

Ο αντικειμενικός στόχος είναι

$\max \sum_{ik} p_i y_{ik}$, $\forall$ i, j, k. (1)

όπου $\sum_{ik} p_i y_{ik}$ οι συνολικές αποδοχές

Περιορισμοί

$\sum_{ijk} c_{ij} x_{ijk} \leq c_{max}$ (2)

όπου $\sum_{ijk} c_{ij} x_{ijk}$ το συνολικό κόστος και c_{max} το μέγιστο κόστος

$\sum_{i} x_{ijk} = y_{jk}$ $\forall$ j, k. (3)

$\sum_{j} x_{ijk} = y_{ik}$ $\forall$ i, k. (4)

Οι περιορισμοί (3) & (4) είναι γνωστοί ως περιορισμοί ανάθεσης. Εξασφαλίζουν ‘ότι όσες φορές το όχημα k επισκεφτεί τον κόμβο i τόσες φορές θα πάει καταφθάσει και θα αναχωρήσει από αυτόν

$$u_{ik} - u_{jk} + Nx_{ijk} \leq N - 1, \gamma\iota\alpha\ i \neq j{:}\ i = 2{,}3 \ldots N, j = 2{,}3 \ldots N, k = 1, \ldots K, u_{ik}, u_{jk} \geq 0 \quad (5)$$

Ο περιορισμός (5) είναι εξασφαλίζει την μη ύπαρξη υποδιαδρομών στη λύση

$$\sum_i a_i y_{ik} \leq b \quad \forall k \qquad (6)$$

Ο περιορισμός (6) εξασφαλίζει να μην ξεπεραστεί η χωρητικότητα του κάθε οχήματος

$$\sum_{\kappa} y_{ik} \begin{cases} k, & i = 1 \\ binary & i - 2, \ldots, n \end{cases} \qquad (7)$$

Ο περιορισμός (7) εξασφαλίζει την χρησιμοποίηση όλων των οχημάτων στην λύση και ότι κάθε κόμβος (εκτός από τον κεντρικό) θα επισκεφτεί από το πολύ από ένα όχημα.

$$\begin{cases} \mathrm{x_{ijk}} = 0 \text{ ή } 1 \\ \mathrm{y_{ik}} = 0 \text{ ή } 1, \gamma\iota\alpha\ i \neq 0 \end{cases} \qquad (8)$$

i, j = 1, 2, …, N ,όπου N ο αριθμός των κόμβων

k= 1, 2, …, K, όπου K ο αριθμός των οχημάτων

The Prize-Collecting VRPP

Ονοματολογία

p_i αποδοχές από την εξυπηρέτηση του κόμβου i

b = χωρητικότητα (σε βάρος ή όγκο του κάθε οχήματος).

a_i = το μέγεθος της παράδοσης στον κόμβο i, μετρημένο στην ίδια μονάδα μέτρησης με την χωρητικότητα των οχημάτων

c_{ij} = το κόστος της απευθείας διαδρομής από τον κόμβο i στον κόμβο j (υποθέτουμε $c_{ij} \geq 0$ και $c_{ij}=c_{ji}$ για όλα τα ij)

N ο συνολικός αριθμός των κόμβων

οι μεταβλητές απόφασης είναι:

y_{ik}: 1 αν το όχημα k επισκέπτεται τον κόμβο i, αλλιώς 0.

x_{ijk}: 1 αν το όχημα k κάνει την απευθείας διαδρομή από τον κόμβο i στον κόμβο j, αλλιώς 0.

Ο αντικειμενικός στόχος είναι

$$\min \sum_{ijk} c_{ij} x_{ijk}\ , \ \forall \text{ i, j, k.} \quad (1)$$

όπου $\sum_{ijk} c_{ij} x_{ijk}$ το συνολικό κόστος

Περιορισμοί

$$\sum_{ik} p_i y_{ik} \geq p_{max} \quad (2)$$

όπου $\sum_{ik} p_i y_{ik}$ οι συνολικές αποδοχές και p_{max} οι ελάχιστες αποδοχές

$$\sum_i x_{ijk} = y_{jk} \ \forall \text{ j, k.} \quad (3)$$

$$\sum_j x_{ijk} = y_{ik} \quad \forall \text{ i, k.} \quad (4)$$

Οι περιορισμοί (3) & (4) είναι γνωστοί ως περιορισμοί ανάθεσης. Εξασφαλίζουν ‘ότι όσες φορές το όχημα k επισκεφτεί τον κόμβο i τόσες φορές θα πάει καταφθάσει και θα αναχωρήσει από αυτόν

$$u_{ik} - u_{jk} + Nx_{ijk} \leq N - 1, \gamma\iota\alpha\ i \neq j: i = 2,3 \ldots N, j = 2,3 \ldots N, k = 1, \ldots K, u_{ik}, u_{jk} \geq 0 \quad (5)$$

Ο περιορισμός (5) εξασφαλίζει την μη ύπαρξη υποδιαδρομών στη λύση

$$\textstyle\sum_i \ a_i y_{ik} \leq b \quad \forall k \qquad (6)$$

Ο περιορισμός (6) εξασφαλίζει να μην ξεπεραστεί η χωρητικότητα του κάθε οχήματος

$$\sum_{\kappa} y_{ik} \begin{cases} k, & i = 1 \\ binary & i - 2, \ldots, n \end{cases} \qquad (7)$$

Ο περιορισμός (7) εξασφαλίζει την χρησιμοποίηση όλων των οχημάτων στην λύση και ότι κάθε κόμβος (εκτός από τον κεντρικό) θα επισκεφτεί από το πολύ από ένα όχημα.

$$\begin{cases} \mathrm{x_{ijk}} = 0 \text{ ή } 1 \\ \mathrm{y_{ik}} = 0 \text{ ή } 1, \gamma\iota\alpha\ i \neq 0 \end{cases} \qquad (8)$$

i, j = 1, 2, …, N ,όπου N ο αριθμός των κόμβων

k= 1, 2, …, K, όπου K ο αριθμός των οχημάτων

Διαδικασία επίλυσης του VRPP

Για την βελτιστοποίηση χρησιμοποιήθηκε το πρόγραμμα LINGO της εταιρίας LINDO. Το LINGO είναι ένα εργαλείο βελτιστοποίησης για γραμμικά, μη γραμμικά και ακέραια συστήματα, σε περιβάλλον windows.

Ο υπολογιστής που χρησιμοποιήθηκε είναι ένας mobile AMD Sempron 3000+ με 512mb μνήμη με λογισμικό windows XP της Microsoft. Οι δοκιμές που έγιναν, μερικές από τις οποίες θα παρουσιαστούν στην συνέχεια, έγιναν με αληθινά ή αληθοφανή δεδομένα. Για παράδειγμα για κάθε πόλη η ζήτησή της (σε προϊόν) επιλέχθηκε ως ανάλογη του πληθυσμού της και οι αποστάσεις μεταξύ των πόλεων είναι οι πραγματικές.

Η προετοιμασία των δεδομένων έγινε στο excel του MS Office το οποίο το Lingo έχει την δυνατότητα να αναγνωρίζει. Επίσης έγινε μια προσπάθεια να γίνει ανάλυση των αποτελεσμάτων τόσο για τον υπολογιστικό χρόνο που απαιτούσαν όσο και στην εξήγηση αυτών.

Οι δοκιμές-τρεξίματα ήταν το πιο χρονοβόρο μέρος της παρούσας μελέτης. Συνολικά έγιναν πάνω από 60 δοκιμές, κάθε μία από τις οποίες διήρκεσε από λίγα δευτερόλεπτα έως και 48 ώρες.

Τέλος, για τα παραδείγματα που θα παρουσιαστούν στην συνέχεια τα αποτελέσματα παρουσιάζονται σε γραφική μορφή έτσι ώστε να είναι κατανοητά στον αναγνώστη. Η γραφική μορφή, αυτών, έγινε σε περιβάλλον Autocad.

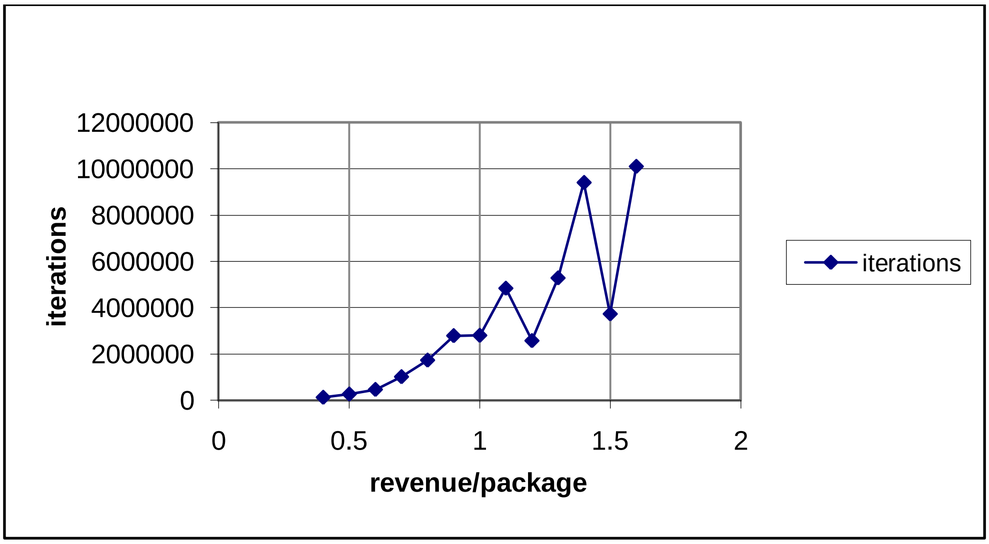
ΑΝΑΛΥΣΗ ΕΥΑΙΣΘΗΣΙΑΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΕΜΠΕΙΡΙΑ

Κατά την διάρκεια της ανάλυσης ευαισθησίας έγιναν πολυάριθμα τρεξίματα του προγράμματος. Τα διάφορα δεδομένα που χρησιμοποιήθηκαν κάθε φορά δεν ήταν τυχαία, αλλά άλλαζαν βάση κάποιου συγκεκριμένου μοτίβου. Έτσι, κρίνεται σκόπιμο να γίνει μία μικρή αναφορά στα αποτελέσματα και ένας σύντομος σχολιασμός τους.

Ως δεδομένα χρησιμοποιήθηκε μία υποθετική διανομή κάποιου υποθετικού προϊόν. Ως κόμβοι χρησιμοποιήθηκαν 15 πόλεις της Ισπανίας, με κεντρικό κόμβο την Μαδρίτη, και τα κόστη υπολογίστηκαν βάση των πραγματικών οδικών αποστάσεων τους. Η ζήτηση σε προϊόν κάθε πόλης επιλέχθηκε ως ανάλογη του πληθυσμού της και στις περισσότερες χρησιμοποιήθηκαν 3 οχήματα με συνολική χωρητικότητα κοντά στην ολική ζήτηση. Τα δεδομένα υπάρχουν αναλυτικά στο παράρτημα.

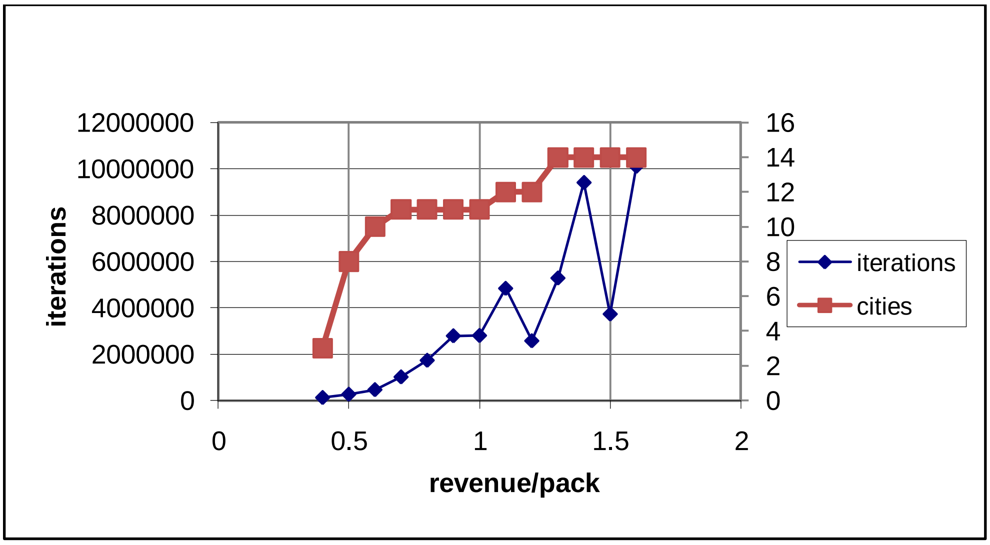
The Profitable Vehicle Routing Problem with Profits

Όσο αφορά το (PVRPP), αρχικά, έγιναν δοκιμές κρατώντας σταθερά τα δεδομένα και αλλάζοντας τις απολαβές ανά πακέτο



διάγραμμα 1 επαναλήψεις του Lingo προς αποδοχές ανά πακέτο για το PVRPP

Παρατηρώντας το διάγραμμα 1 είναι εμφανές πως αυξάνοντας το αντίτιμο που παίρνει η εταιρία ανά πακέτο, το πρόγραμμα καθυστερεί να δώσει λύση. Ο χρόνος που κάνει το πρόγραμμα δείχνει να σταθεροποιείται για μεγάλα αντίτιμα, με κάποιες διακυμάνσεις. Μια λογική εξήγηση σε αυτό είναι ότι για χαμηλά αντίτιμα οι λύσεις είναι σχετικά απλές. Για μικρό αντίτιμο, δηλαδή, η εταιρεία σίγουρα θα έχει ζημιά και έτσι κάθε όχημα θα επισκεφτεί μία το πολύ πόλη ώστε η ζημιά να ελαχιστοποιηθεί. Όσο αυξάνεται το αντίτιμο οι πόλεις που επισκέπτεται κάθε όχημα θα αυξάνονται και έτσι η λύση θα γίνεται όλο και πιο πολύπλοκη, φτάνοντας σε ένα σημείο όπου όλες οι πόλεις θα επισκέπτονται και η περαιτέρω αύξηση του αντίτιμου δεν θα οδηγεί σε πιο πολύπλοκη λύση. Αυτό μπορεί να γίνει εμφανές στο διάγραμμα 2.



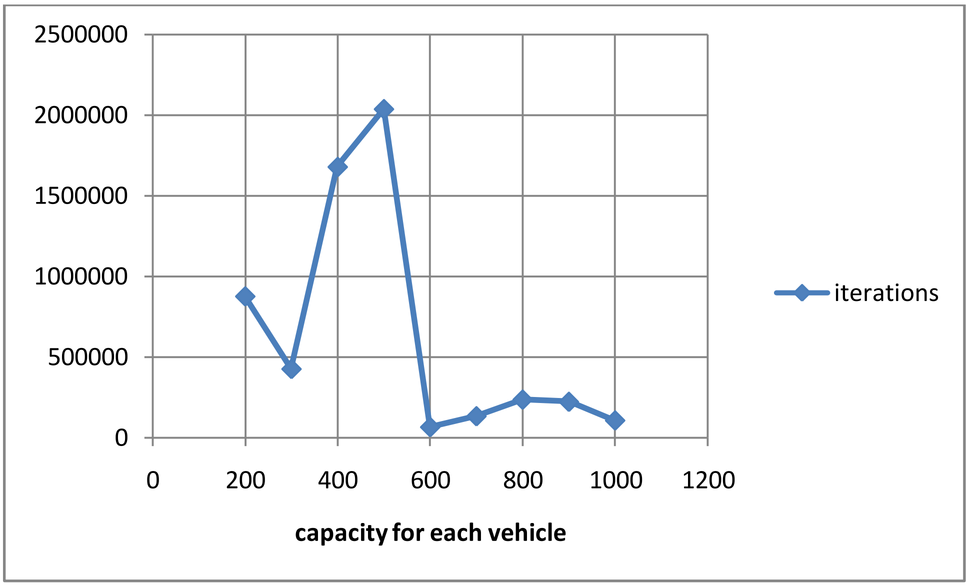
διάγραμμα 2 επαναλήψεις του Lingo και πόλεις που εξυπηρετούνται προς αποδοχές ανά πακέτο για το PVRPP

rev/pack	iterations	revenue	cost	income
0.4	131403	176.4	311.34	-134.94
0.5	272335	379.5	447.54	-68.04
0.6	470482	512.4	496.2	16.2
0.7	1020647	618.1	515.65	102.45
0.8	1736753	706.4	515.65	190.75
0.9	2793169	795.7	515.65	280.05
1	2806090	883	515.65	367.35
1.1	4840226	1037.3	579.59	457.71
1.2	2578062	1131.6	579.59	552.01
1.3	5285179	1301.3	653.89	647.41
1.4	9405317	1401.4	653.89	747.51
1.5	3733208	1501.5	653.89	847.61
1.6	10100571	1601.6	653.89	947.71

πίνακας 1 αποτελέσματα μεταβολής του παράγοντα αποδοχές ανά πακέτο για το PVRPP

Τα στοιχεία του πίνακα 1 δεν είναι και πολύ χρήσιμα στην εταιρεία. Είναι λογικό όσο αυξάνεται το αντίτιμο ανά πακέτο να αυξάνονται τα κέρδη. Το μόνο που μπορούμε να αναφέρουμε είναι ότι υπάρχει ένα ελάχιστο αντίτιμο ανά πακέτο, στην προκειμένη περίπτωση 0.6, για το οποίο η εταιρεία παρουσιάζει κέρδη.

Στη συνέχεια έγιναν δοκιμές διατηρώνας σταθερά τα δεδομένα και αλλάζοντας την μεταφορική ικανότητα των οχημάτων



διάγραμμα 3 επαναλήψεις του Lingo προς χωρητικότητα κάθε οχήματος για το PVRPP

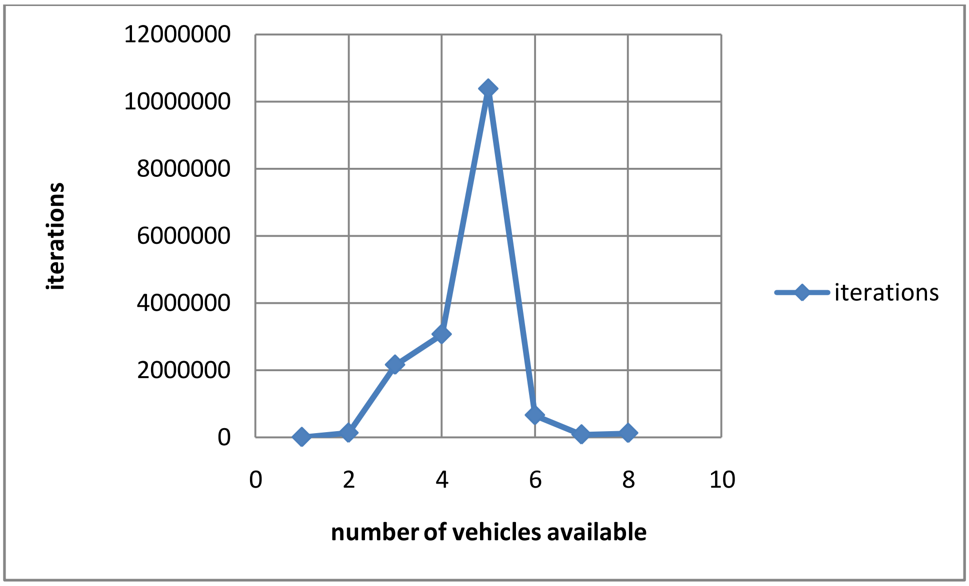
Παρατηρώντας το διάγραμμα 3 γίνεται αντιληπτό πως ο χρόνος που κάνει το πρόγραμμα για να δώσει αποτελέσματα ελαχιστοποιείται για οχήματα με μεγάλη χωρητικότητα. Μία λογική εξήγηση σε αυτό είναι ότι σε αυτές τις περιπτώσεις, ο περιορισμός των συνολικών πακέτων ανά όχημα παύει να υφίσταται και έτσι οδηγούμαστε σε απλούστερο πρόβλημα.

capacity	iterations	revenue	cost	income
200	876992	420.8	418.36	2.44
300	426561	492	472.53	19.47
400	1679642	731.2	533.02	198.18
500	2038248	754.4	547.62	206.78
600	66272	754.4	513.55	240.85
700	133565	754.4	513.55	240.85
800	236828	754.4	513.55	240.85
900	223028	754.4	513.55	240.85
1000	107449	754.4	513.55	240.85

πίνακας 2 αποτελέσματα μεταβολής του παράγοντα χωρητικότητα κάθε οχήματος για το PVRPP

Τα στοιχεία του πίνακα 2 μπορεί να φανούν αρκετά χρήσιμα σε μία εταιρεία. Βάση αυτών μπορεί κάποιος να αποφασίσει εάν τα υπάρχοντα οχήματα είναι επαρκή ή όχι. Είναι λογικό όσο αυξάνεται η χωρητικότητα των οχημάτων, να αυξάνεται το κέρδος της εταιρείας. Βέβαια, όπως είναι εμφανές, υπάρχει ένα μέγιστο όριο χωρητικότητας όπου περεταίρω αύξησή της δεν οδηγεί σε μεγαλύτερα κέρδη, αφού από ένα σημείο και μετά η χωρητικότητα των οχημάτων πάντα υπερκαλύπτει την ζήτηση. Στη συγκεκριμένη περίπτωση είναι 600. Μάλιστα οχήματα χωρητικότητας μεγαλύτερης των τετρακοσίων δίνουν ικανοποιητικά κέρδη (82% του μέγιστου δυνατού)

Τέλος για το PVRPP έγιναν δοκιμές αλλάζοντας των αριθμό των οχημάτων με όλα τα υπόλοιπα δεδομένα να διατηρουνται σταθερά



διάγραμμα 4 επαναλήψεις του Lingo προς εύρος διαθέσιμων οχημάτων για το PVRPP

Όσο αυξάνεται ο αριθμός των οχημάτων, αυξάνεται και η πολυπλοκότητα με το λογικό αντίκτυπο σε υπολογιστικό χρόνο. Όσο όμως ο αριθμός των οχημάτων φτάνει κοντά στον αριθμό των πόλεων το πρόβλημα απλοποιείται. Στην περίπτωση, μάλιστα, που τα οχήματα είναι ίσα με τις πόλεις η λύση είναι μοναδική, αφού κάθε όχημα αναγκαστικά επισκέπτεται και από μία πόλη. Το παραπάνω διάγραμμα, λοιπόν, αντικατοπτρίζει ακριβώς αυτό.

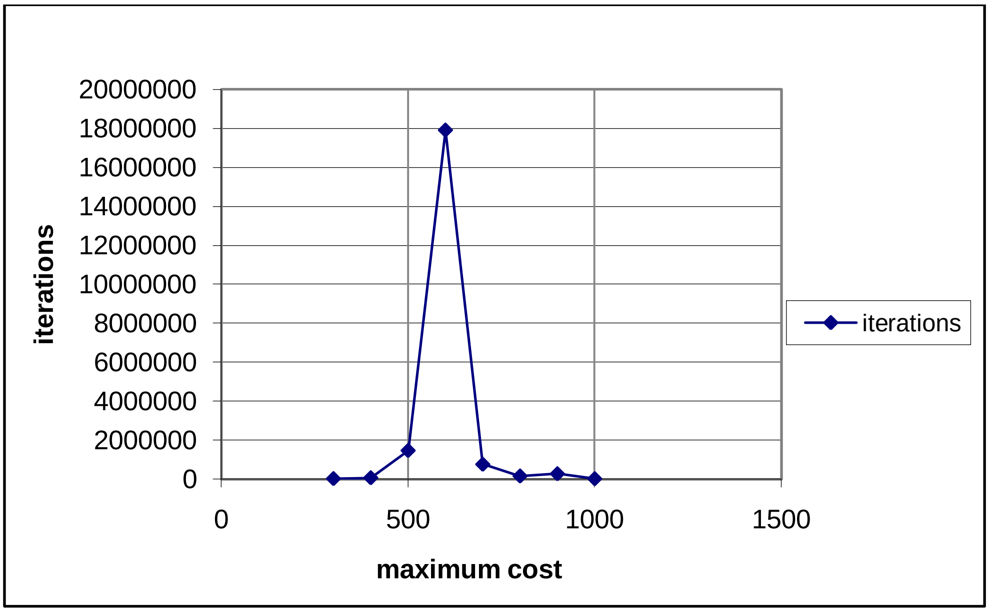
ships	iterations	revenue	cost	income
1	227	262	147.34	115
2	127404	504.8	341.92	163
3	2157078	684	510.09	174
4	3066673	754.4	619.87	135
5	10386284	754.4	689.36	65
6	655319	754.4	758.86	-4
7	77154	754.4	843.66	-89
8	122735	754.4	932.61	-178

πίνακας 3 αποτελέσματα μεταβολής του παράγοντα χωρητικότητα κάθε οχήματος για το PVRPP

Τα στοιχεία του πίνακα 3 είναι πολύ σημαντικά σε μια εταιρεία. Βάση αυτών μπορεί να πάρει κάποιος την απόφαση πόσα οχήματα πρέπει να χρησιμοποιηθούν, αν υπάρχει έλλειψη οχημάτων ή αν χρησιμοποιούνται υπεραρκετά.

The Team Orienteering Problem

Η περίπτωση μεγιστοποίησης αποδοχών με ένα μέγιστο κόστος (TOP) είναι αρκετά διαφορετική από την προηγούμενη περίπτωση. Στην περίπτωση αυτή υπάρχει μία παράμετρος, το μέγιστο κόστος, που δεν υπήρχε προηγουμένως και η πολυπλοκότητα του προβλήματος εξαρτάται άμεσα από αυτήν όπως είναι εμφανές στο διάγραμμα 5.



διάγραμμα 5 επαναλήψεις του Lingo προς μέγιστο κόστος για το TOP

Για το λόγο αυτό, αν και έγιναν αρκετές δοκιμές με διάφορα δεδομένα, τα αποτελέσματα των οποίων, όπως και ο υπολογιστικός χρόνος, θα παρατεθούν στην συνέχεια, δεν κρίνεται σκόπιμο να γίνει αναφορά στη διακύμανση του υπολογιστικού χρόνου και στην πολυπλοκότητα του κάθε προβλήματος, αφού δοκιμές με διαφορετικά μέγιστα κόστη θα οδηγούσαν σε τελείως διαφορετικά αποτελέσματα και συμπεράσματα.

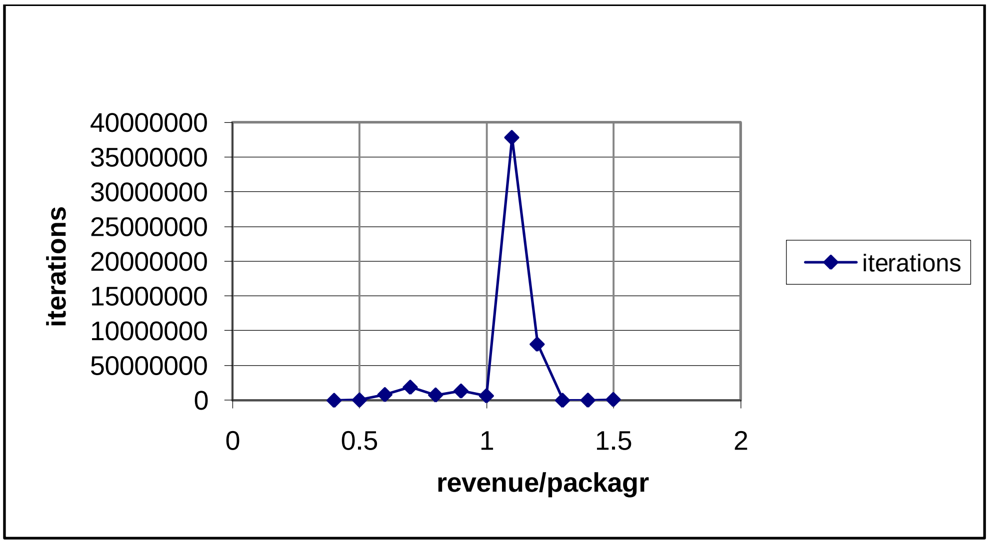
Στην προκειμένη περίπτωση που τα δεδομένα του προβλήματος διατηρήθηκαν σταθερά, ενώ το μέγιστο κόστος έλαβε τιμές από 300 έως 1000 και τα αποτελέσματα είναι εμφανή στον πίνακα 4:

maxcost	iterations	revenue	cost	profit
300	19227	182.4	290.47	-108.07
400	62271	504.8	387.78	117.02
500	1460578	683.2	496.2	187
600	17900311	754.4	599.48	154.92
700	753420	800.8	699.77	101.03
800	152192	800.8	788.72	12.08
900	268000	800.8	892.47	-91.67
1000	13561	800.8	994.74	-193.94

πίνακας 4 αποτελέσματα μεταβολής του παράγοντα προς μέγιστο κόστος για το TOP

Τα αποτελέσματα, που φαίνονται στον πίνακα 4, ήταν σχετικά αναμενόμενα. Τα κόστη είναι αρκετά κοντά στο μέγιστο κόστος έτσι ώστε οι αποδοχές να είναι όσο το δυνατόν μεγαλύτερες. Επίσης είναι εμφανές ότι η αύξηση του μέγιστου κόστους από μία τιμή και μετά δεν οδηγεί σε μεγαλύτερες αποδοχές. Στην περίπτωση αυτή, αύξηση του μέγιστου κόστους πάνω από την τιμή του 700 δεν οδηγεί σε μεγαλύτερες αποδοχές. Αν παρατηρήσουμε τη διακύμανση του κέρδους, γίνεται κατανοητό ότι η τιμή του μέγιστου κόστους είναι κάτι που χρειάζεται σοβαρή μελέτη έτσι ώστε τα αποτελέσματα να οδηγούν σε κέρδος και να είναι χρήσιμα στην εταιρεία.

Δοκιμές έγιναν επίσης διατηρώντας σταθερά τα δεδομένα και αλλάζοντας τις τιμές αποδοχών ανά πακέτο. Το μέγιστο κόστος στην κάθε περίπτωση ήταν κατά 50 μονάδες μεγαλύτερο της βέλτιστης λύσης του προηγούμενου προβλήματος (PVRPP). Τα αποτελέσματα μαζί με τους υπολογιστικούς χρόνους είναι εμφανή στο διάγραμμα 6.



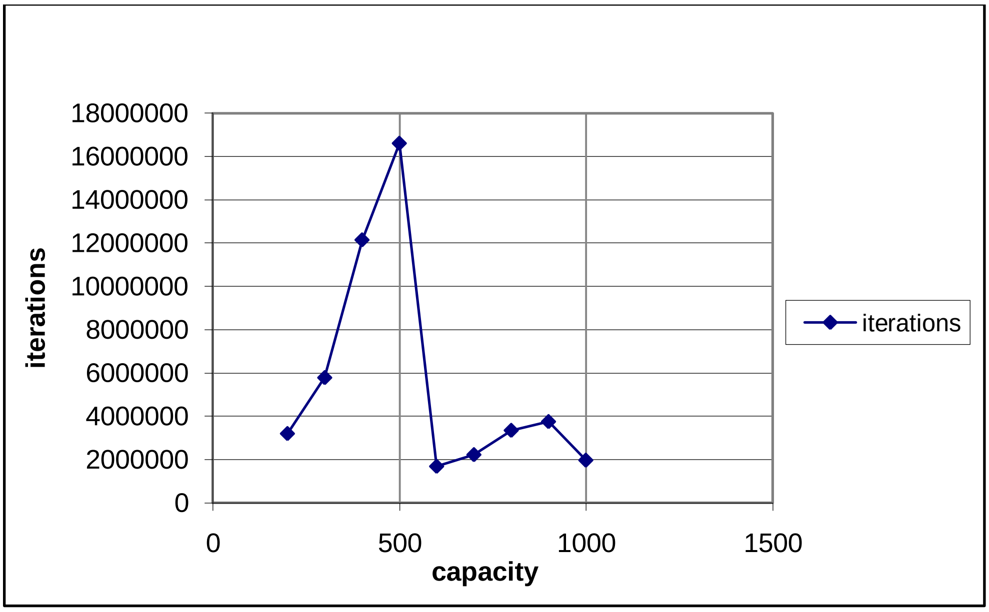
διάγραμμα 6 επαναλήψεις του Lingo προς αποδοχές ανά πακέτο για το TOP

rev/pack	top max cost	iterations	revenue	cost	income
0.4	361.34	36665	200.4	353.03	-152.63
0.5	497.54	597374	427	496.2	-69.2
0.6	546.2	8327488	531.6	544.84	-13.24
0.7	565.65	18939220	640.5	551.78	88.72
0.8	565.65	7785587	732	564.29	167.71
0.9	565.65	13456352	823.5	564.29	259.21
1	565.65	6565273	915	551.78	363.22
1.1	629.59	378223192	1049.4	626.1	423.3
1.2	629.59	80803452	1144.8	626.1	518.7
1.3	703.89	80340	1301.3	698.35	602.95
1.4	703.89	281118	1401.4	695.61	705.79
1.5	703.89	1259413	1501.5	699.75	801.75

πίνακας 5 αποτελέσματα μεταβολής του παράγοντα αποδοχές ανά πακέτο για το TOP

Είναι αξιοσημείωτο το ότι για μια τιμή αποδοχών ανά πακέτο ο υπολογιστικός χρόνος είναι πολλαπλάσιος των υπολοίπων. Η εξήγηση αυτού χρειάζεται περαιτέρω μελέτη που δεν κρίθηκε σκόπιμο να γίνει στο πλαίσιο της παρούσας εργασίας. Τα συγκεκριμένα αποτελέσματα δεν είναι ιδιαίτερα χρήσιμα στην εταιρεία, αφού είναι λογικό αύξηση των αποδοχών ανά πακέτο να σημαίνει μεγαλύτερες αποδοχές και κέρδη.

Τέλος έγιναν δοκιμές κρατώντας σταθερά τα δεδομένα και αλλάζοντας την χωρητικότητα των οχημάτων. Το μέγιστος κόστος λήφθηκε, όπως και στην προηγούμενη περίπτωση, κατά 50 μονάδες μεγαλύτερο της βέλτιστης λύσης του προηγούμενου προβλήματος (PVRPP). Τα αποτελέσματα φαίνονται στο διάγραμμα 7.



διάγραμμα 7 επαναλήψεις του Lingo προς χωρητικότητα κάθε οχήματος για το TOP

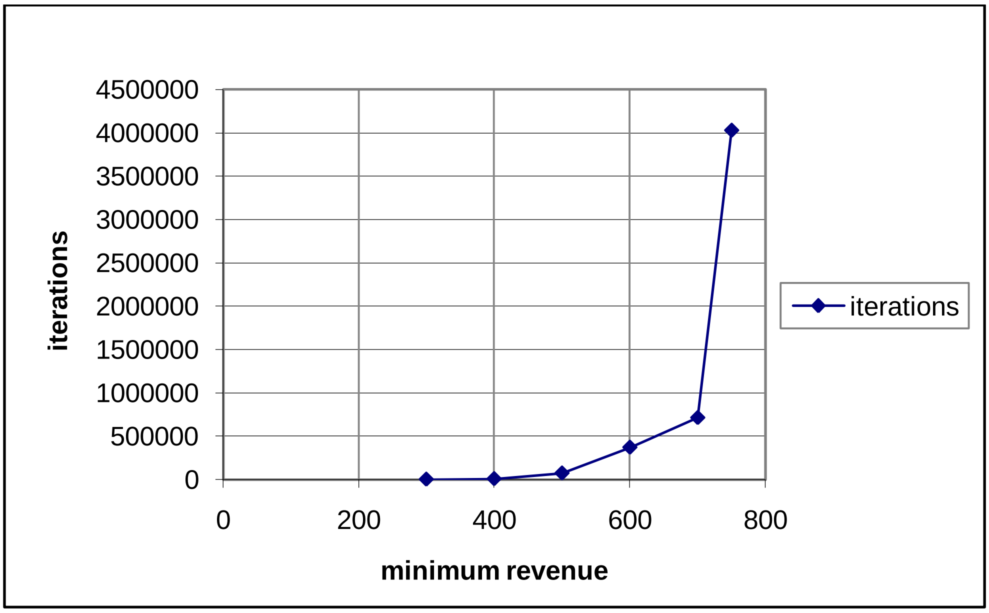
capacity	maxcost	iterations	revenue	cost	income
200	468.36	3188128	446.4	454.5	-8.1
300	522.53	5772484	516	519.49	-3.49
400	583.02	12137612	755.2	572.6	182.6
500	597.62	16600081	778.4	594.82	183.58
600	563.55	1674813	778.4	560.51	217.89
700	563.55	2214501	778.4	560.51	217.89
800	563.55	3337258	778.4	560.51	217.89
900	563.55	3741776	778.4	560.51	217.89
1000	563.55	1957963	778.4	560.51	217.89

πίνακας 6 αποτελέσματα μεταβολής του παράγοντα χωρητικότητας κάθε οχήματος για το TOP

Τα στοιχεία του πίνακα 6 μπορούν να οδηγήσουν σε χρήσιμα συμπεράσματα για την εταιρεία. Για τα συγκεκριμένα δεδομένα, χωρητικότητα οχημάτων μεγαλύτερη από 500 οδηγεί στις μέγιστες αποδοχές, ενώ ακόμα και στην τιμή 400 οι αποδοχές είναι αρκετά κοντά στις μέγιστες δυνατές. Επίσης, αύξηση της χωρητικότητας σε τιμή ανώτερη από 600 δεν οδηγεί σε αύξηση των κερδών. Αξίζει να σημειωθεί ότι αυτό συμφωνεί με τα αποτελέσματα του προβλήματος (PVRPP). Έχοντας υπόψη τα στοιχεία αυτά, μια εταιρεία μπορεί να κρίνει αν η χωρητικότητα των ήδη υπάρχων οχημάτων είναι ικανοποιητική ή αν χρειάζεται να γίνει ανανέωση.

The Prize-Collecting VRPP

Η περίπτωση ελαχιστοποίησης κόστους με μία ελάχιστη αποδοχή (PCVRPP) μοιάζει αρκετά με την προηγούμενη περίπτωση (TOP). Όπως και πριν, στην περίπτωση αυτή υπάρχει μία παράμετρος (οι ελάχιστες αποδοχές) η οποία επηρεάζει άμεσα την πολυπλοκότητα του προβλήματος. Αυτό γίνεται εμφανές στο διάγραμμα 8.



διάγραμμα 8 επαναλήψεις του Lingo προς ελάχιστες αποδοχές για το PC-VRPP

Για το λόγο αυτό, αν και έγιναν αρκετές δοκιμές με διάφορα δεδομένα, των οποίων τα αποτελέσματα και ο υπολογιστικός χρόνος θα παρατεθούν στην συνέχεια, δεν κρίνεται σκόπιμο να γίνει αναφορά στη διακύμανση του υπολογιστικού χρόνου και στην πολυπλοκότητα του κάθε προβλήματος, αφού δοκιμές με διαφορετικές ελάχιστες αποδοχές θα οδηγούσαν σε τελείως διαφορετικά αποτελέσματα και συμπεράσματα.

Από το διάγραμμα 8 είναι εμφανές ότι όσο αυξάνονται οι ελάχιστες αποδοχές το πρόβλημα γίνεται πιο πολύπλοκο και ο υπολογιστικός χρόνος μεγαλύτερος. Αυτό είναι λογικό αν σκεφτεί κανείς ότι όσο αυξάνονται οι ελάχιστες αποδοχές, αυξάνεται και ο αριθμός των πόλεων που πρέπει να επισκεφτούν και έτσι το πρόβλημα γίνεται πιο πολύπλοκο.

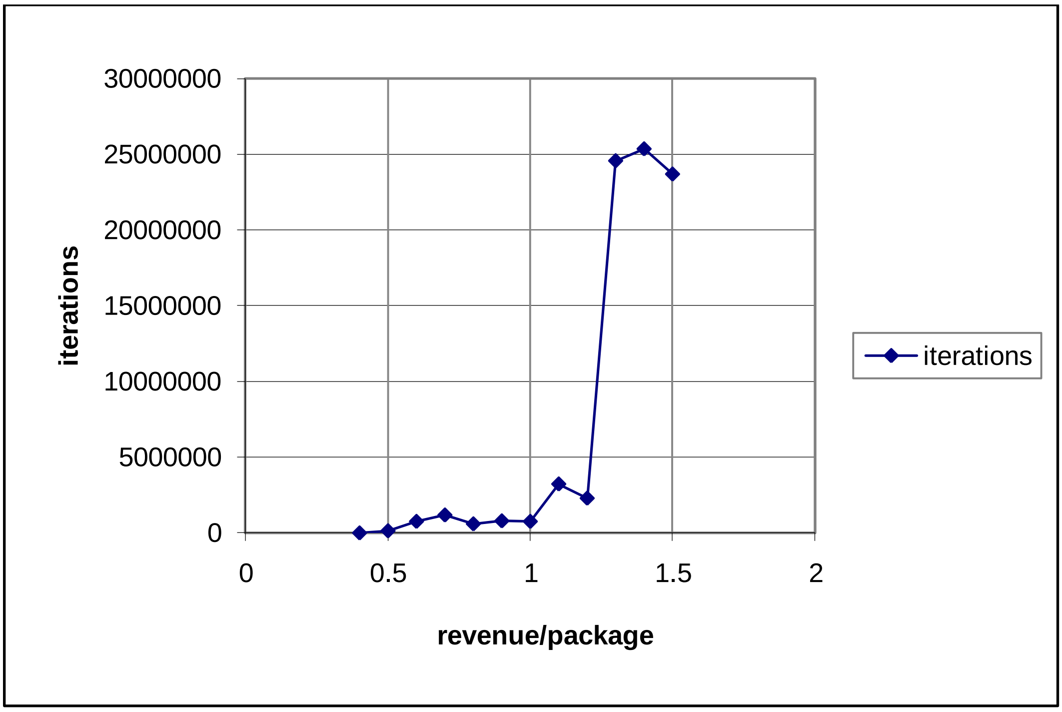
Στην προκειμένη περίπτωση τα δεδομένα του προβλήματος διατηρήθηκαν σταθερά, ενώ το μέγιστο κόστος έλαβε τιμές από 300 έως 750. Σε κάθε περίπτωση οι αποδοχές, το κόστος, το κέρδος και οι επαναλήψεις που έκανε το LINGO για να βγάλει αποτελέσματα, ήταν:

min revenue	iterations	revenue	cost	profit
300	966	352.8	311.34	41.46
400	6967	400.8	353.03	47.77
500	72008	504.8	387.78	117.02
600	370144	607.2	447.54	159.66
700	712979	706.4	515.65	190.75
750	4033499	754.4	579.59	174.81

πίνακας 7 αποτελέσματα μεταβολής του παράγοντα ελάχιστες αποδοχές για το PC-VRPP

Τα αποτελέσματα του πίνακα 8 ήταν σχετικά αναμενόμενα. Οι αποδοχές είναι αρκετά κοντά στις ελάχιστες δυνατές, έτσι ώστε το κόστος να είναι όσο το δυνατόν μικρότερο.

Δοκιμές έγιναν επίσης διατηρώντας σταθερά τα δεδομένα και αλλάζοντας τις τιμές αποδοχών ανά πακέτο. Οι ελάχιστες αποδοχές στην κάθε περίπτωση ήταν κατά 50 μονάδες μικρότερες της βέλτιστης λύσης του προβλήματος (PVRPP). Τα αποτελέσματα μαζί με τους υπολογιστικούς χρόνους είναι:



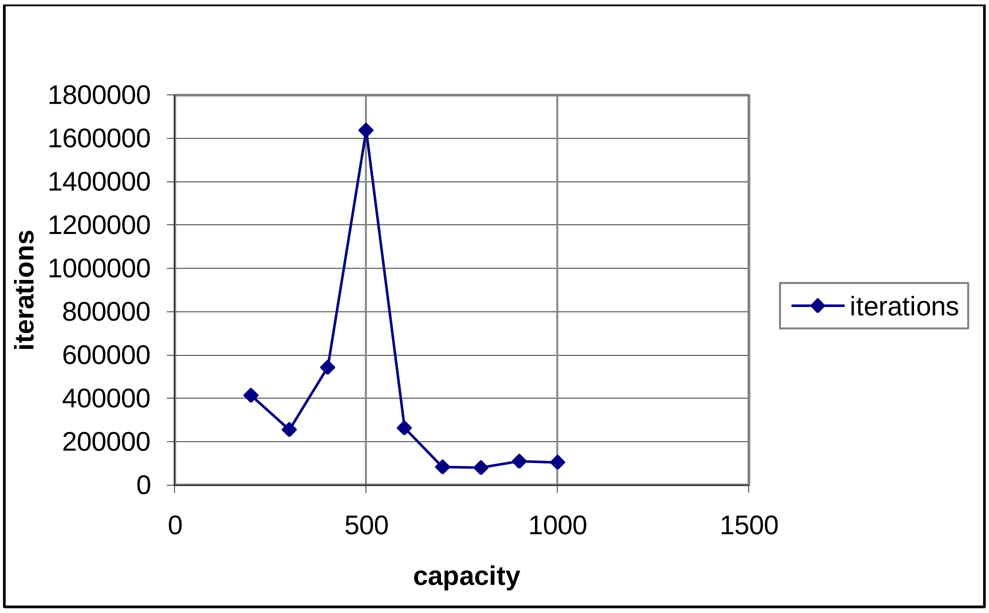
διάγραμμα 9 επαναλήψεις του Lingo προς αποδοχές ανά πακέτο για το PC-VRPP

rev/pack	min revenue	iterations	revenue	cost	profit
0.4	126.4	3689	176.4	311.34	-134.94
0.5	329.5	145287	329.5	410.01	-80.51
0.6	462.4	766326	472.8	466.99	5.81
0.7	568.1	1189129	569.8	489.26	80.54
0.8	656.4	601478	660.8	490.64	170.16
0.9	745.7	800296	768.6	496.2	272.4
1	833	764788	854	496.2	357.8
1.1	987.3	3237292	993.3	550.4	442.9
1.2	1081.6	2298811	1083.6	550.4	533.2
1.3	1251.3	24581021	1263.6	634.44	629.16
1.4	1351.4	25367100	1360.8	634.44	726.36
1.5	1451.5	23701720	1458	634.44	823.56

πίνακας 8 αποτελέσματα μεταβολής του παράγοντα αποδοχές ανά πακέτο για το PC-VRPP

Τα συγκεκριμένα αποτελέσματα δεν είναι ιδιαίτερα χρήσιμα στην εταιρεία αφού είναι λογικό, αύξηση των αποδοχών ανά πακέτο να σημαίνει μεγαλύτερες αποδοχές και κέρδη.

Τέλος έγιναν δοκιμές κρατώντας σταθερά τα δεδομένα και αλλάζοντας την χωρητικότητα των οχημάτων. Οι ελάχιστες αποδοχές λήφθηκαν, όπως και στην προηγούμενη περίπτωση, κατά 50 μονάδες μικρότερες της βέλτιστης λύσης του προβλήματος (PVRP). Τα αποτελέσματα είναι:



διάγραμμα 10 επαναλήψεις του Lingo προς χωρητικότητα κάθε οχήματος για το PC-VRPP

capacity	min revenue	iterations	revenue	cost	profit
200	370.8	412526	388.8	404.45	-15.65
300	442	254252	444	430.84	13.16
400	681.2	541445	683.2	491.33	191.87
500	704.4	1635777	706.4	505.93	200.47
600	704.4	262087	706.4	471.86	234.54
700	704.4	81738	706.4	471.86	234.54
800	704.4	78579	706.4	471.86	234.54
900	704.4	108022	706.4	471.86	234.54
1000	704.4	103426	706.4	471.86	234.54

πίνακας 9 αποτελέσματα μεταβολής του παράγοντα χωρητικότητα κάθε οχήματος για το PC-VRPP

Είναι σχεδόν αδύνατο από τα στοιχεία του πίνακα 9 κάποιος να συσχετίσει το ελάχιστο κόστος με την χωρητικότητα των οχημάτων. Ο παράγοντας των ελάχιστων αποδοχών επηρεάζει άμεσα το ελάχιστο κόστος ώστε να βγει κάποιο χρήσιμο συμπέρασμα. Έτσι ο σχολιασμός θα γίνει μόνο με το κέρδος.

Τα στοιχεία του πίνακα 8 μπορούν να οδηγήσουν σε χρήσιμα συμπεράσματα για την εταιρεία. Για τα συγκεκριμένα δεδομένα, χωρητικότητα οχημάτων μεγαλύτερη από 600 οδηγεί στο μέγιστο δυνατό κέρδος, ενώ ακόμα και στην τιμή 400 το κέρδος είναι αρκετά κοντά στο μέγιστο δυνατό. Επίσης, αύξηση της χωρητικότητας σε τιμή ανώτερη από 600 δεν οδηγεί σε αύξηση των κερδών. Αξίζει να σημειωθεί ότι αυτό συμφωνεί με τα αποτελέσματα και των δύο προηγούμενων προβλημάτων (PVRP και TOP). Έτσι η απόφαση που πρέπει να πάρει η εταιρεία για το αν η χωρητικότητα των ήδη υπάρχων οχημάτων είναι ικανοποιητική ή αν χρειάζεται να γίνει ανανέωση, μπορεί να γίνει με βάση των αποτελεσμάτων, οποιουδήποτε από τα 3 είδη προβλήματος.

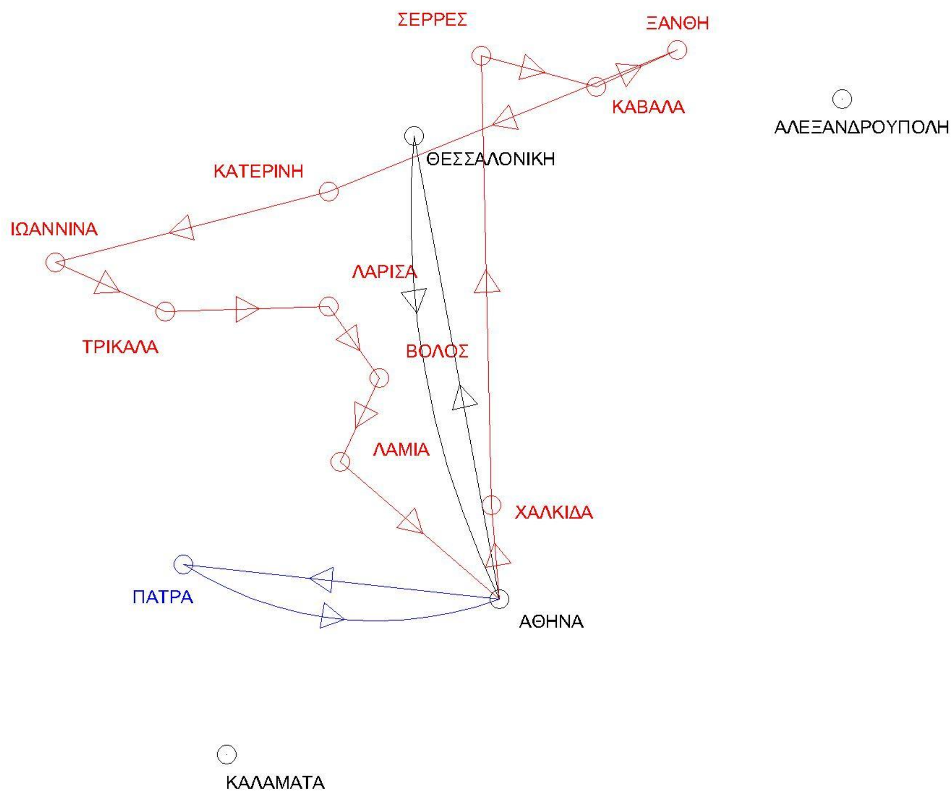
ΠΑΡΑΔΕΙΓΜΑΤΑ

Κατά την διάρκεια της παρούσας μελέτης έγιναν μερικές δοκιμές πάνω σε μια υποθετική διανομή προϊόντων σε 15 πόλεις τις Ελλάδας. Σε ένα υποθετικό σενάριο μία εταιρεία με έδρα την Αθήνα παράγει ένα προϊόν και εξετάζει την διάθεση του στην επαρχία. Μετά από έρευνα αγοράς έγινε γνωστή η ζήτηση του προϊόντος της στις 14 μεγαλύτερες πόλεις της Ελλάδας.

Λόγω έλλειψης πραγματικών δεδομένων ως κόστος επιλέχθηκε η οδική απόσταση των πόλεων πολλαπλασιασμένη με μία τυπική κατανάλωση ενός επαγγελματικού βαν (9,2 λίτρα ανά 100χλμ). Η συνολική ζήτηση σε προϊόν όλων των πόλεων επιλέχθηκε 1000 πακέτα και κατανεμήθηκε στις πόλεις αναλογικά ως προς τον πληθυσμό τους. Χρησιμοποιήθηκαν 3 οχήματα με χωρητικότητα, ανάλογα με την περίπτωση, από 450 έως 500 πακέτα και ανάλογα με την περίπτωση οι απολαβές ανά πακέτο ήταν από 0,8 έως 1,3 ευρώ. Τα δεδομένα για κάθε παράδειγμα υπάρχουν αναλυτικά στο παράρτημα.

The Profitable Vehicle Routing Problem with Profits

Στην περίπτωση αυτή οι διαδρομές θα είναι αυτές που βελτιστοποιούν το κέρδος. Στο πρώτο παράδειγμα οι απολαβές ανά πακέτο επιλέχθηκαν ίσες με 0,8 ενώ η χωρητικότητα κάθε οχήματος 450 πακέτα. Τα αποτελέσματα στη γραφική τους μορφή είναι

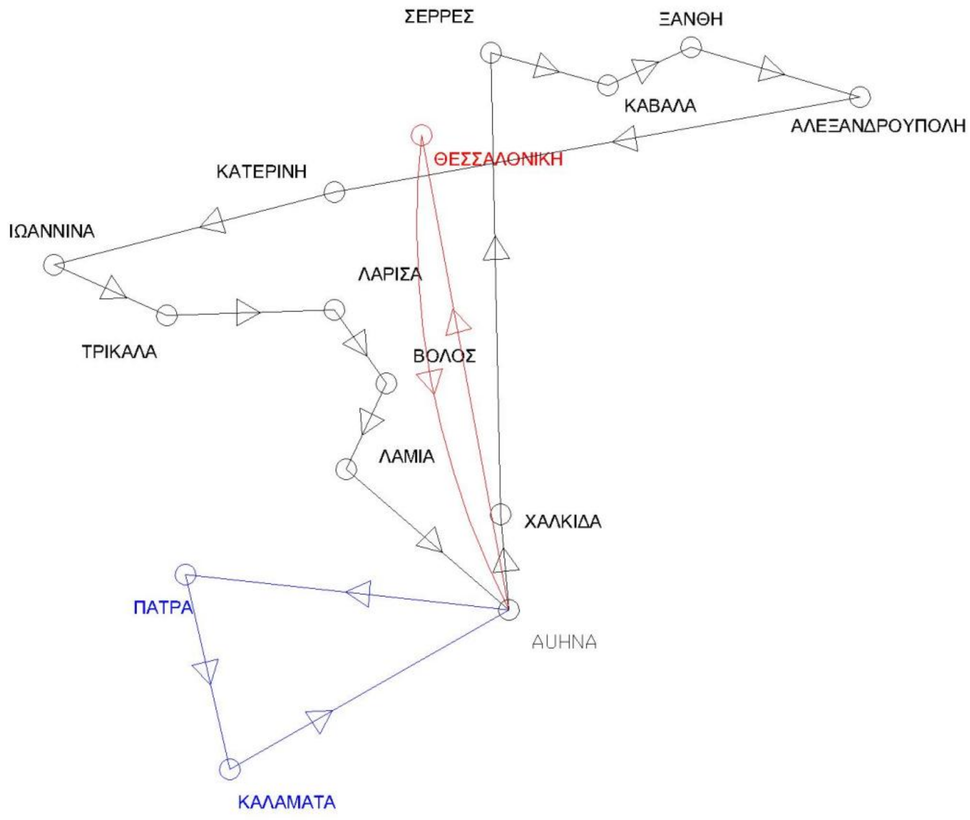


σχήμα 3 αποτελέσματα 1^{ου} παραδείγματος για το PVRPP

Στο σχήμα 3 βλέπουμε πως αποκλείονται από την διαδρομή η Καλαμάτα και η Αλεξανδρούπολη. Αυτό συμβαίνει επειδή η βελτιστοποίηση του κέρδους απαιτεί διαδρομές με πόλεις που αποφέρουν λιγότερα έσοδα από το κόστος να αποκλείονται από την διαδρομή. Για το λόγο του αληθές, θα μπορούσε κάποιος να πει ότι εφόσον η Ξάνθη συμπεριλαμβάνεται στις διαδρομές, θα έπρεπε να συμπεριληφθεί και η Αλεξανδρούπολη. Αυτό φαίνεται εύλογο αφού επίσκεψη της Αλεξανδρούπολης θα απέφερε 20.8 ευρώ και το κόστος από Ξάνθη είναι 11.224. Φυσικά αυτό είναι λάθος . Τα αποτελέσματα δείχνουν την διαδρομή Ξάνθη-Κατερίνη ως βέλτιστη. Αυτή έχει κέρδος 23.2(έσοδα από την επίσκεψη στην Κατερίνη)-26.772(κόστος διαδρομής)=-

3.572 ευρώ. Η διαδρομή, για να συμπεριληφθεί και η Αλεξανδρούπολη, θα έπρεπε να είναι Ξάνθη-Αλεξανδρούπολη-Κατερίνη και θα απέφερε κέρδη 20.8(έσοδα από την επίσκεψη στην Αλεξανδρούπολη)+23.2(έσοδα από την επίσκεψη στην Κατερίνη)-(11.224+37.812) (κόστος διαδρομής)=-5.036<-3.572. Θα ζημίωνε δηλαδή την εταιρεία με 1.464 ευρώ.

Στο επόμενο παράδειγμα αυξήσαμε τις αποδοχές ανά πακέτο κατά 0,5. Τα αποτελέσματα σε γραφική μορφή είναι:



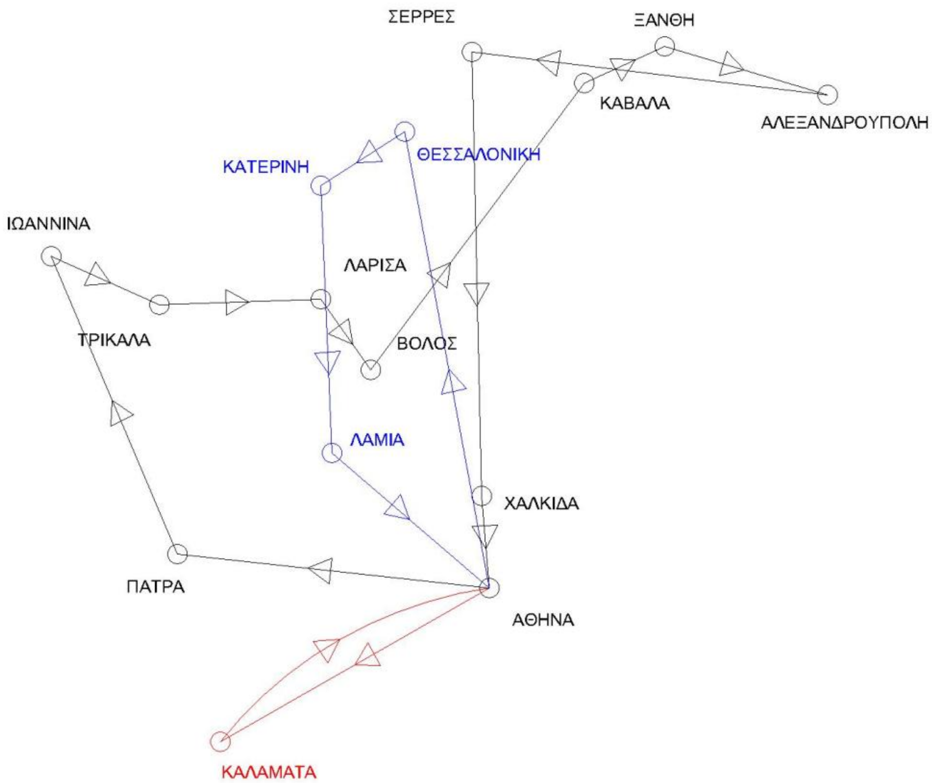
σχήμα 4 αποτελέσματα $2^{ου}$ παραδείγματος για το PVRPP

Τα αποτελέσματα δεν παρουσίασαν κάτι μη αναμενόμενο. Λόγω των υψηλών αποδοχών ανά πακέτο και αφού η χωρητικότητα των οχημάτων το επιτρέπει, όλες οι πόλεις εξυπηρετούνται από τα οχήματα.

Επίσης για όποιον γνωρίζει την γεωγραφία της Ελλάδας οι διαδρομές που ακολουθούν τα οχήματα είναι λογικές. Η σειρά, δηλαδή, με την οποία επισκέπτονται τα οχήματα τις πόλεις είναι η ίδια με αυτήν με που θα ακολουθούσε κάποιος αν έπρεπε να επισκεφτεί αυτές τις πόλεις, χωρίς να κάνει άσκοπες διαδρομές. Αυτό βέβαια ήταν αναμενόμενο αφού η ελαχιστοποίηση του κόστους το επιβάλλει.

Εμφανείς είναι και οι περιορισμοί της χωρητικότητας. Ιδιαίτερα στην περίπτωση του οχήματος που επισκέπτεται μόνο την Θεσσαλονίκη. Στην περίπτωση αυτή αν και υπάρχουν πόλεις, κοντά στην Θεσσαλονίκη, που θα ήταν λογικό να τις επισκεφτεί το όχημα αυτό δεν συμβαίνει, αφού η χωρητικότητα του οχήματος δεν επαρκεί.

Στην επόμενη περίπτωση αυξήσαμε την χωρητικότητα του κάθε οχήματος κατά 50 πακέτα, ενώ οι αποδοχές ανά πακέτο διατηρήθηκαν στο 1,3. Τα αποτελέσματα είναι τα εξής:



σχήμα 5 αποτελέσματα 3ου παραδείγματος για το PVRPP

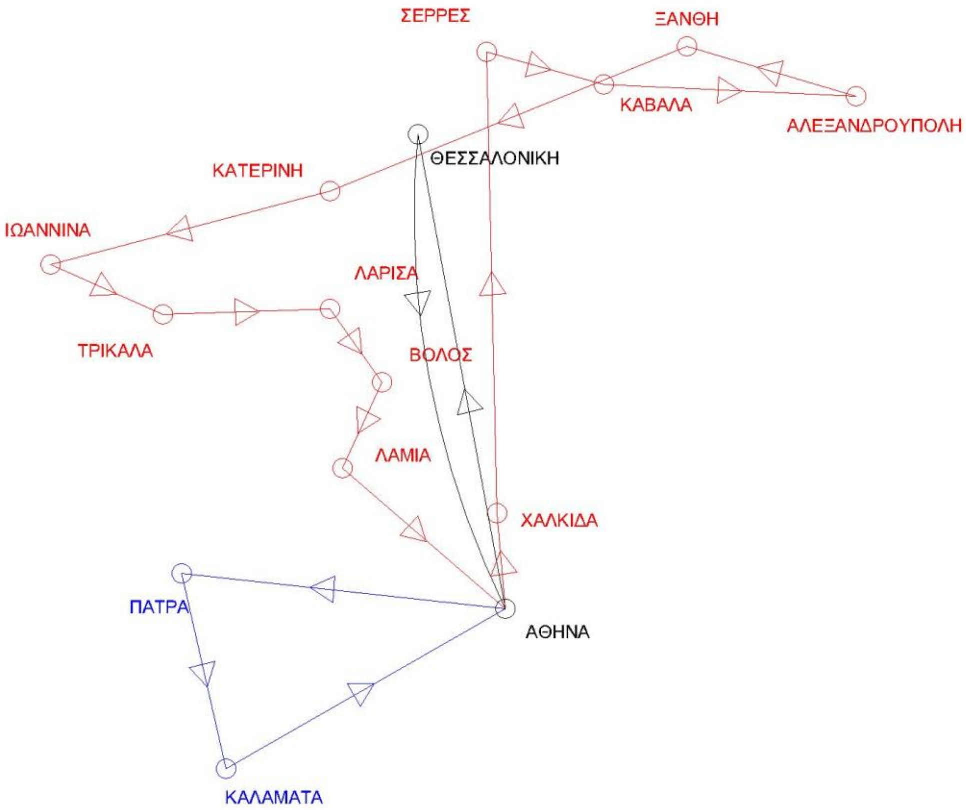
Όπως ήταν αναμενόμενο λόγω των υψηλών αποδοχών ανά πακέτο και στην περίπτωση αυτή, όλες οι πόλεις επισκέπτονται. Βλέπουμε ότι οι διαδρομές είναι λίγο διαφορετικές. Λόγω της αύξησης της χωρητικότητας, το όχημα που επισκέπτεται την Θεσσαλονίκη κάνει μεγαλύτερη διαδρομή και επισκέπτεται 2 πόλεις που βρίσκονται κοντά στην διαδρομή Αθήνα-Θες/νίκη. Αν παρατηρήσουμε, προσεκτικά, όμως το σχήμα 5 θα δούμε ότι υπάρχει ένα όχημα που αναλώνεται στην διαδρομή Αθήνα-Καλαμάτα (χωρίς να επισκέπτεται άλλες πόλεις). Αυτό συμβαίνει επειδή το πρόγραμμα είναι αναγκασμένο να χρησιμοποιήσει και τα 3 οχήματα. Αν σκεφτεί κάποιος πως η χρησιμοποίηση ενός οχήματος επιφέρει και επιπλέων κόστη όπως το μεροκάματο του οδηγού γίνεται εμφανές ότι στην περίπτωση αυτή αυτό το όχημα είναι πλεονάζων και δεν θα έπρεπε να χρησιμοποιηθεί, αφού η χωρητικότητα των 2 υπόλοιπων είναι επαρκής. Παρόλα αυτά η λύση είναι καλύτερη από την προηγούμενη. Φυσικά, οι αποδοχές είναι οι βέλτιστες δυνατές και στις 2 λύσεις, αφού όλες οι πόλεις επισκέπτονται. Το κόστος όμως στην περίπτωση αυτή είναι μικρότερο της προηγούμενης (306.82< 320.16). Αυτό ήταν αναμενόμενο αφού οι μεγαλύτερες χωρητικότητες οχημάτων δίνουν την δυνατότητα να ακολουθηθούν διαδρομές με μικρότερα κόστη.

The Team Orienteering Problem

Στην περίπτωση αυτή η μεγιστοποίηση των αποδοχών επιτυγχάνεται με τα οχήματα να επισκέπτονται όσο το δυνατόν περισσότερες πόλεις. Βέβαια, αυτό περιορίζεται από το μέγιστο κόστος το οποίο δεν πρέπει να ξεπεραστεί.

Αυτό είναι εμφανές στα αποτελέσματα. Στην περίπτωση αυτή, μάλιστα, τα οχήματα επισκέφτηκαν όλες τις πόλεις, αφού το μέγιστο κόστος το επέτρεψε. Πόλεις όπως Αλεξανδρούπολη και Καλαμάτα αντίθετα με την περίπτωση (PVRPP), μιλώντας πάντα για ίδιες αποδοχές ανά πακέτο, τώρα επισκέπτονται από τα οχήματα και αυξάνουν τις αποδοχές φτάνοντάς τες στο μέγιστο δυνατό. Στο συγκεκριμένο παράδειγμα η χωρητικότητα του κάθε οχήματος επιλέχθηκε 450 πακέτα και οι αποδοχές ανά πακέτο ίσες με 0,8 ευρώ.

Τα αποτελέσματα είναι:

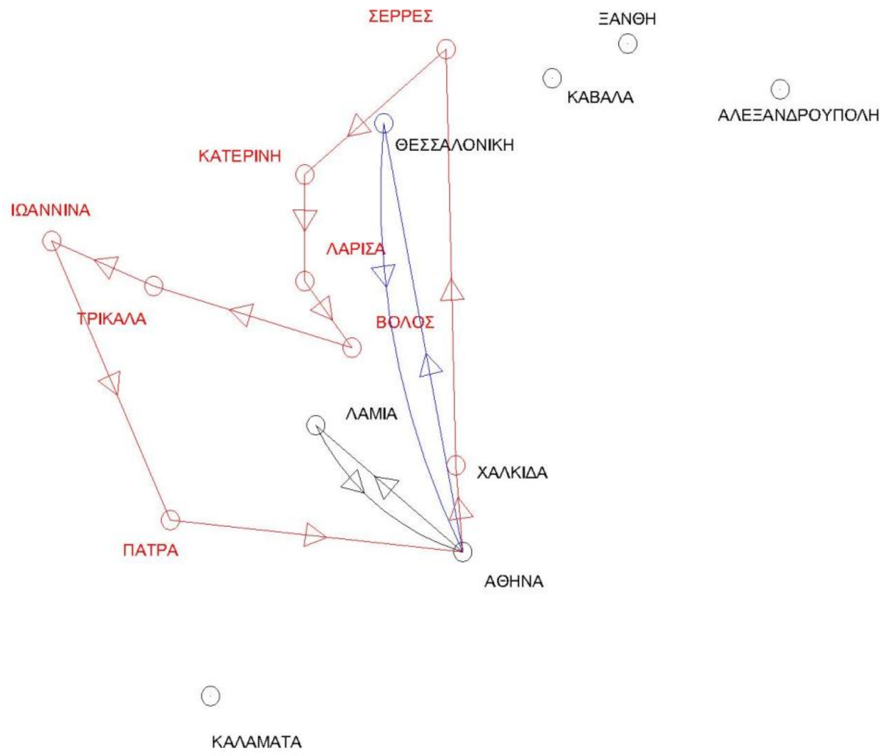


σχήμα 6 αποτελέσματα παραδείγματος για το TOP

The Prize-Collecting VRPP

Στην περίπτωση αυτή η ελαχιστοποίηση του κόστους επιτυγχάνεται με τα οχήματα να επισκέπτονται όσο το δυνατόν λιγότερες πόλεις και με τις διαδρομές να είναι όσο το δυνατόν πιο μικρές. Βέβαια, πρέπει να εξασφαλίζονται οι ελάχιστες αποδοχές. Έτσι υπάρχει και ένας ελάχιστος αριθμός πόλεων που πρέπει να επισκεφτούν τα οχήματα.

Στο παράδειγμα η χωρητικότητα επιλέχθηκε ίση με 450 πακέτα και οι αποδοχές ανά πακέτο ίσες με 0,8 ευρώ.



σχήμα 7 αποτελέσματα παραδείγματος για το PC-VRPP

Αυτό είναι εμφανές από τα αποτελέσματα. Υπάρχουν πόλεις που θα μπορούσαν να αυξήσουν το κέρδος (κάτι το οποίο είναι εμφανές από τα προηγούμενα παραδείγματα) αλλά στην περίπτωση αυτή δεν επισκέπτονται από τα οχήματα κρατώντας, έτσι, το κόστος χαμηλό.

Παρατηρώντας τα αποτελέσματα είναι εμφανές πως στην συγκεκριμένη περίπτωση το 3^{ο} όχημα (που κάνει την διαδρομή Αθήνα-Λαμία) είναι πλεονάζων και το κόστος θα ήταν μικρότερο αν χρησιμοποιούσαμε μόνο 2 οχήματα.

ΕΠΕΚΤΑΣΗ ΜΟΝΤΕΛΟΥ – ΤΟ FLEET SELECTIVE VRPP (FS-VRPP)

Κατά την διάρκεια των δοκιμαστικών τρεξιμάτων του VRPP ανέκυψε ένα πρόβλημα. Ας πούμε πως μια εταιρεία αντιμετωπίζει ένα πρόβλημα δρομολόγησης οχημάτων με κέρδη. Ουσιαστικά, πρέπει να διαλέξει μέσα από ένα πλήθος πελατών, αυτούς που μεγιστοποιούν τα κέρδη της. Πριν βάλει σε εφαρμογή το μαθηματικό μοντέλο του VRPP , η εταιρεία, πρέπει να πάρει μία απόφαση, πόσα οχήματα θα χρησιμοποιήσει. Ακριβώς επειδή οι πελάτες που τελικά θα εξυπηρετηθούν, και επομένως ο όγκος του μεταφερόμενου φορτίου, δεν είναι γνωστοί στο σημείο αυτό, η απόφαση αυτή είναι πολύ δύσκολη και μπορεί να οδηγήσει σε μη βέλτιστη λύση. Για τον λόγο αυτό κάναμε μία προσπάθεια αντιμετώπισης του προβλήματος επιλογής βέλτιστου στόλου και δρομολόγησης οχημάτων ως ένα ενιαίο πρόβλημα.

Το πρόβλημα, αφού δεν βρήκαμε κάτι στην διεθνή βιβλιογραφία, το ονομάσαμε Fleet Selective Vehicle Routing Problem with Profits. Στο FS-VRPP το ζητούμενο είναι ο καθορισμός των διαδρομών που πρέπει να ακολουθήσει ένας στόλος οχημάτων, με έδρα έναν κεντρικό κόμβο, με σκοπό την εξυπηρέτηση ενός συνόλου γεωγραφικά διασκορπισμένων πελατών, έτσι ώστε το κέρδος να είναι μέγιστο. Η διαφορά με την προηγούμενη περίπτωση (VRPP) είναι ότι δεν είναι αναγκαίο όλα τα οχήματα να χρησιμοποιηθούν στην λύση, αν όμως χρησιμοποιηθούν υπάρχει ένα αρχικό κόστος που πρέπει να πληρωθεί.

Αναλυτικά, υπάρχει ένας στόλος μη όμοιων οχημάτων που έχει ως έδρα έναν κεντρικό κόμβο, ο οποίος ουσιαστικά είναι μία πολύ μεγάλη αποθήκη κάποιου προϊόντος . Γύρω από τον κεντρικό κόμβο υπάρχουν διασκορπισμένοι πελάτες (κόμβοι). Οι πελάτες έχουν συγκεκριμένη ζήτηση για το προϊόν του κεντρικού κόμβου. Τα έσοδα από την εξυπηρέτηση κάποιου κόμβου είναι επίσης γνωστά. Αν κάποιο όχημα χρησιμοποιηθεί στην λύση υπάρχει ένα αρχικό κόστος που πρέπει να πληρωθεί (κόστος οδηγού, ασφάλειας κτλ)

- Δεν είναι αναγκαίο όλα τα οχήματα να χρησιμοποιηθούν στην λύση, το ίδιο ισχύει και για τους πελάτες.
- Το κάθε όχημα έχει την δική του χωρητικότητα, κατά βάση διαφορετική των υπολοίπων
- Κάθε πελάτης εξυπηρετείται το πολύ μια φορά.
- Όλα τα οχήματα ξεκινούν από τον κεντρικό κόμβο και στο τέλος επιστρέφουν σε αυτόν.
- Το κόστος διαδρομής μεταξύ δύο κόμβων είναι γνωστό και διαφορετικό για κάθε όχημα
- Ένα όχημα δεν μπορεί να εξυπηρετήσει μόνο ένα ποσοστό της ζήτησης κάποιου πελάτη, στην περίπτωση που τον εξυπηρετεί παραδίδει το 100% της ζήτησής του.

Το ζητούμενο είναι οι διαδρομές που πρέπει να ακολουθήσουν τα οχήματα. Στο PVRPP οι διαδρομές είναι αυτές που μεγιστοποιούν το κέρδος, στην περίπτωση του TOP μεγιστοποιούν τα έσοδα ενώ στο PCVRPP ελαχιστοποιούν το κόστος.

ΤΑ ΜΑΘΗΜΑΤΙΚΑ ΜΟΝΤΕΛΑ

The Fleet Selective Profitable Vehicle Routing Problem With Profits

Ονοματολογία

p_i αποδοχές από την εξυπηρέτηση του κόμβου i

b_k = χωρητικότητα (σε βάρος ή όγκο του κάθε οχήματος).

a_i = το μέγεθος της παράδοσης στον κόμβο i, μετρημένο στην ίδια μονάδα μέτρησης με την χωρητικότητα των οχημάτων

c_{ijk} = το κόστος της απευθείας διαδρομής από τον κόμβο i στον κόμβο j για το όχημα k (υποθέτουμε $c_{ikj} \geq 0$ και $c_{ijk}=c_{jik}$ για όλα τα ijk)

N ο συνολικός αριθμός κόμβων

οι μεταβλητές απόφασης είναι:

y_{ik}: 1 αν το όχημα k επισκέπτεται τον κόμβο i, αλλιώς 0.

x_{ijk}: 1 αν το όχημα k κάνει την απευθείας διαδρομή από τον κόμβο i στον κόμβο j, αλλιώς 0.

y_{ik}: 1 αν το όχημα k επισκέπτεται τον κόμβο i, αλλιώς 0.

g_k: το αρχικό κόστος εκκίνησης του οχήματος k

Ο αντικειμενικός στόχος είναι

$$max \sum_{ik} p_i y_{ik} - \sum_{ijk} c_{ijk} x_{ijk} - \sum_k g_k y_1 , \forall i,j,k \qquad (1)$$

όπου $\sum_{ik} p_i y_{ik}$ οι συνολικές αποδοχές

$\sum_{ijk} c_{ijk} x_{ijk}$ το συνολικό κόστος

και $\sum_k g_k y_1$ το συνολικό αρχικό κόστος για όλα τα οχήματα που θα χρησιμοποιηθούν

Περιορισμοί

$$\sum_i x_{ijk} = y_{jk} \ \forall \text{ j, k.} \quad (2)$$

$$\sum_j x_{ijk} = y_{ik} \quad \forall \text{ i, k.} \ (3)$$

Οι περιορισμοί (2) & (3) είναι γνωστοί ως περιορισμοί ανάθεσης. Εξασφαλίζουν ότι όσες φορές το όχημα k επισκεφτεί τον κόμβο i τόσες φορές θα πάει καταφθάσει και θα αναχωρήσει από αυτόν

$$u_{ik} - u_{jk} + Nx_{ijk} \leq N - 1, \gamma\iota\alpha\ i \neq j: i = 2{,}3 \ldots N, j = 2{,}3 \ldots N, k = 1, \ldots K, u_{ik}, u_{jk} \geq 0 \ (4)$$

Ο περιορισμός (4) εξασφαλίζει την μη ύπαρξη υποδιαδρομών στην λύση. Απόδειξή του έχει γίνει στην περίπτωση του VRPP.

$$\sum_i a_i y_{ik} \leq b_k \quad \forall \text{ k.} \qquad (5)$$

Ο περιορισμός (5) εξασφαλίζει να μην ξεπεραστεί η χωρητικότητα του κάθε οχήματος

$$\sum_k y_{ik} = binary, i = 2 \ldots . n \quad (6)$$

Ο περιορισμός (6) εξασφαλίζει την χρησιμοποίηση όλων των οχημάτων στην λύση και ότι κάθε κόμβος (εκτός από τον κεντρικό) θα επισκεφτεί από το πολύ από ένα όχημα.

$$\begin{cases} \mathrm{x_{ijk}} = 0 \text{ ή } 1 \\ \mathrm{y_{ik}} = 0 \text{ ή } 1, \gamma\iota\alpha\ i \neq 1 \end{cases} \quad (7)$$

i, j = 1, 2, …, N ,όπου N ο αριθμός των κόμβων

k= 1, 2, …, K, όπου K ο αριθμός των οχημάτων

The Team Orienteering Problem

Ονοματολογία

p_i αποδοχές από την εξυπηρέτηση του κόμβου i

b_k = χωρητικότητα (σε βάρος ή όγκο του κάθε οχήματος).

a_i = το μέγεθος της παράδοσης στον κόμβο i, μετρημένο στην ίδια μονάδα μέτρησης με την χωρητικότητα των οχημάτων

c_{ijk} = το κόστος της απευθείας διαδρομής από τον κόμβο i στον κόμβο j για το όχημα k (υποθέτουμε $c_{ikj} \geq 0$ και $c_{ijk}=c_{jik}$ για όλα τα ijk)

N ο συνολικός αριθμός κόμβων

οι μεταβλητές απόφασης είναι:

y_{ik}: 1 αν το όχημα k επισκέπτεται τον κόμβο i, αλλιώς 0.

x_{ijk}: 1 αν το όχημα k κάνει την απευθείας διαδρομή από τον κόμβο i στον κόμβο j, αλλιώς 0.

y_{ik}: 1 αν το όχημα k επισκέπτεται τον κόμβο i, αλλιώς 0.

g_k: το αρχικό κόστος εκκίνησης του οχήματος k

οι μεταβλητές απόφασης είναι:

y_{ik}: 1 αν το όχημα k επισκέπτεται τον κόμβο i, αλλιώς 0.

x_{ijk}: 1 αν το όχημα k κάνει την απευθείας διαδρομή από τον κόμβο i στον κόμβο j, αλλιώς 0.

Ο αντικειμενικός στόχος είναι

$\max \sum_{ik} p_i y_{ik}$, $\forall$ i, j, k. (1)

όπου $\sum_{ik} p_i y_{ik}$ οι συνολικές αποδοχές

Περιορισμοί

$$\sum_{ijk} c_{ijk} x_{ijk} + \sum_k g_k y_1 , \forall i, j, k \leq c_{max} \qquad (2)$$

όπου $\sum_{ijk} c_{ijk}x_{ijk}$ το συνολικό κόστος και c_{max} το μέγιστο κόστος και $\sum_k g_k y_1$ το συνολικό αρχικό κόστος για όλα τα οχήματα που θα χρησιμοποιηθούν

$$\sum_i x_{ijk} = y_{jk} \quad \forall \text{ j, k.} \quad (3)$$

$$\sum_j x_{ijk} = y_{ik} \quad \forall \text{ i, k.} \quad (4)$$

Οι περιορισμοί (3) & (4) είναι γνωστοί ως περιορισμοί ανάθεσης. Εξασφαλίζουν ότι όσες φορές το όχημα k επισκεφτεί τον κόμβο i τόσες φορές θα πάει καταφθάσει και θα αναχωρήσει από αυτόν

$$u_{ik} - u_{jk} + Nx_{ijk} \leq N - 1, \gamma\iota\alpha\ i \neq j: i = 2{,}3 \dots N, j = 2{,}3 \dots N, k = 1, \dots K, u_{ik}, u_{jk} \geq 0 \quad (5)$$

Ο περιορισμός (5) εξασφαλίζει την μη ύπαρξη υποδιαδρομών στη λύση

$$\sum_i a_i y_{ik} \leq b_k \quad \forall \text{ k.} \quad (6)$$

Ο περιορισμός (6) εξασφαλίζει να μην ξεπεραστεί η χωρητικότητα του κάθε οχήματος

$$\sum_k y_{ik} = binary, i = 2 \dots . n \quad (7)$$

Ο περιορισμός (7) εξασφαλίζει την χρησιμοποίηση όλων των οχημάτων στην λύση και ότι κάθε κόμβος (εκτός από τον κεντρικό) θα επισκεφτεί από το πολύ από ένα όχημα.

$$\begin{cases} \mathrm{x_{ijk}} = 0 \text{ ή } 1 \\ \mathrm{y_{ik}} = 0 \text{ ή } 1,\ \gamma\iota\alpha\ i \neq 1 \end{cases} \quad (8)$$

i, j = 1, 2, …, N ,όπου N ο αριθμός των κόμβων

k= 1, 2, …, K, όπου K ο αριθμός των οχημάτων

The Prize-Collecting VRPP

Ονοματολογία

p_i αποδοχές από την εξυπηρέτηση του κόμβου i

b_k = χωρητικότητα (σε βάρος ή όγκο του κάθε οχήματος).

a_i = το μέγεθος της παράδοσης στον κόμβο i, μετρημένο στην ίδια μονάδα μέτρησης με την χωρητικότητα των οχημάτων

c_{ijk} = το κόστος της απευθείας διαδρομής από τον κόμβο i στον κόμβο j για το όχημα k (υποθέτουμε $c_{ikj} \geq 0$ και $c_{ijk}=c_{jik}$ για όλα τα ijk)

N ο συνολικός αριθμός κόμβων

οι μεταβλητές απόφασης είναι:

y_{ik}: 1 αν το όχημα k επισκέπτεται τον κόμβο i, αλλιώς 0.

x_{ijk}: 1 αν το όχημα k κάνει την απευθείας διαδρομή από τον κόμβο i στον κόμβο j, αλλιώς 0.

y_{ik}: 1 αν το όχημα k επισκέπτεται τον κόμβο i, αλλιώς 0.

g_k: το αρχικό κόστος εκκίνησης του οχήματος k

οι μεταβλητές απόφασης είναι:

y_{ik}: 1 αν το όχημα k επισκέπτεται τον κόμβο i, αλλιώς 0.

x_{ijk}: 1 αν το όχημα k κάνει την απευθείας διαδρομή από τον κόμβο i στον κόμβο j, αλλιώς 0.

Ο αντικειμενικός στόχος είναι

$$min \sum_{ijk} c_{ijk} x_{ijk} + \sum_k g_k y_1 \, , \forall i, j, k$$

όπου $\sum_{ijk} c_{ijk} x_{ijk}$ το συνολικό κόστος

και $\sum_k g_k y_1$ το συνολικό αρχικό κόστος για όλα τα οχήματα που θα χρησιμοποιηθούν

Περιορισμοί

$$\sum_{ik} p_i y_{ik} \geq p_{max} \quad (2)$$

όπου $\sum_{ik} p_i y_{ik}$ οι συνολικές αποδοχές και p_{max} οι ελάχιστες αποδοχές

$$\sum_{i} x_{ijk} = y_{jk} \; \forall \text{ j, k.} \quad (3)$$

$$\sum_j x_{ijk} = y_{ik} \quad \forall \text{ i, k.} \quad (4)$$

Οι περιορισμοί (3) & (4) είναι γνωστοί ως περιορισμοί ανάθεσης. Εξασφαλίζουν ότι όσες φορές το όχημα k επισκεφτεί τον κόμβο i τόσες φορές θα πάει καταφθάσει και θα αναχωρήσει από αυτόν

$$u_{ik} - u_{jk} + Nx_{ijk} \leq N - 1, \gamma\iota\alpha\ i \neq j: i = 2{,}3 \dots N, j = 2{,}3 \dots N, k = 1, \dots K, u_{ik}, u_{jk} \geq 0 \quad (5)$$

Ο περιορισμός (5) εξασφαλίζει την μη ύπαρξη υποδιαδρομών στη λύση

$$\sum_i a_i y_{ik} \leq b_k \quad \forall \text{ k.} \qquad (6)$$

Ο περιορισμός (6) εξασφαλίζει να μην ξεπεραστεί η χωρητικότητα του κάθε οχήματος

$$\textstyle\sum_k y_{ik} = binary, i = 2 \dots . n \quad (7)$$

Ο περιορισμός (7) εξασφαλίζει την χρησιμοποίηση όλων των οχημάτων στην λύση και ότι κάθε κόμβος (εκτός από τον κεντρικό) θα επισκεφτεί από το πολύ από ένα όχημα.

$$\begin{cases} \mathrm{x_{ijk}} = 0 \text{ ή } 1 \\ \mathrm{y_{ik}} = 0 \text{ ή } 1,\ \gamma\iota\alpha\ i \neq 1 \end{cases} \qquad (8)$$

i, j = 1, 2, …, N ,όπου N ο αριθμός των κόμβων

k= 1, 2, …, K, όπου K ο αριθμός των οχημάτων

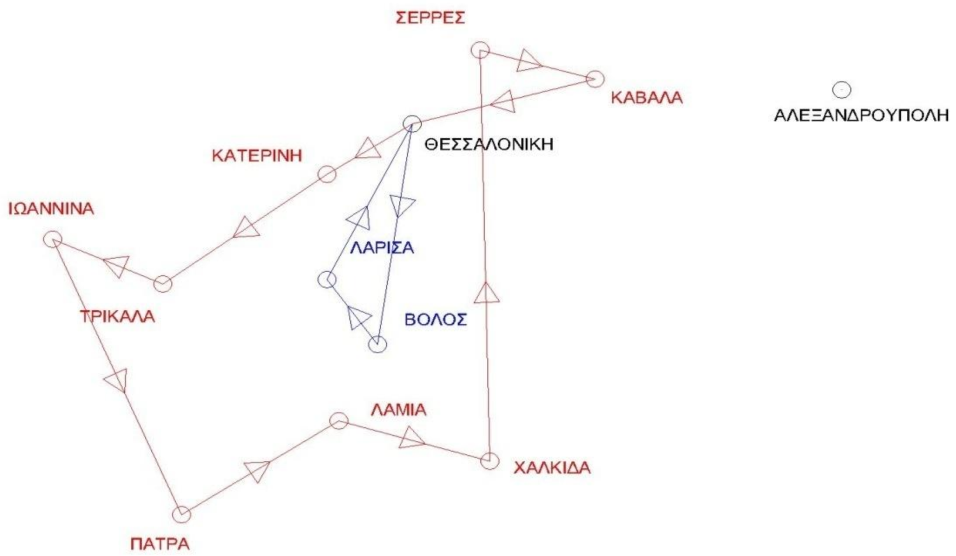
ΠΑΡΑΔΕΙΓΜΑ

Το παράδειγμα που ακολουθεί αφορά μια υποθετική διανομή προϊόντων σε 12 πόλεις τις Ελλάδας με βάση την Θεσσαλονίκη. Ως κόστος επιλέχθηκε η οδική απόσταση των πόλεων πολλαπλασιασμένη με την κατανάλωση του κάθε οχήματος. Επιλέχθηκαν 3 τυχαία επαγγελματικά βαν συνολικής χωρητικότητας 1160 πακέτων η οποία κατανεμήθηκε στα 3 οχήματα ανάλογα την πραγματικής τους. Ως αρχικό κόστος επιλέχθηκε το 1% του κόστος αγοράς του κάθε οχήματος. Η συνολική ζήτηση σε προϊόν όλων των πόλεων επιλέχθηκε 1002 πακέτα και κατανεμήθηκε στις πόλεις αναλογικά ως προς τον πληθυσμό τους. Οι απολαβές ανά πακέτο ήταν από 0,8 ευρώ. Τα στοιχεία του κάθε οχήματος, αναλυτικά, είναι:

Όχημα	αρχικό κόστος	κατανάλωση	χωρητικότητα
1	138.89	10.23l/km	315
2	153.75	9.4l/km	170
3	156.80	11.76l/km	675

πίνακας 10 κύρια χαρακτηριστικά οχημάτων παραδείγματος FS-VRPP

Το παράδειγμα αφορά την περίπτωση του PVRPP και τα αποτελέσματα σε γραφική μορφή είναι:



σχήμα 8 αποτελέσματα παραδείγματος για το FS-VRPP

Εντύπωση κάνει ότι επιλέγονται τα οχήματα που επιλέγονται είναι το 1^{ο} και το 3^{ο}. Το 1^{ο} προφανώς για το μικρό κόστος εκκίνησης και το 3^{ο} για την μεγάλη του χωρητικότητα. Οι δύο αυτοί παράγοντες, στην περίπτωση αυτή, είναι πιο σημαντικοί από την κατανάλωση, αφού το 2^{ο} όχημα που έχει και την μικρότερη κατανάλωση δεν επιλέχθηκε. Το 1^{ο} όχημα κάνει μία αρκετά μικρή διαδρομή (Θεσ/νίκη-Βόλος-Λάρισα) και έτσι η μεγάλη κατανάλωση που έχει δεν επηρεάζει σημαντικά το κόστος, η οποία έτσι και αλλιώς αντισταθμίζεται από το μικρό κόστος εκκίνησης του οχήματος. Το 3^{ο} όχημα επισκέπτεται ένα μεγάλο αριθμό πόλεων εκμεταλλεύοντας έτσι την μεγάλη χωρητικότητά του. Ο συνδυασμός των 2 οχημάτων επιτρέπει τον παροπλισμό του 2^{ου} οχήματος και έτσι το κόστος κρατείται χαμηλά.

ΕΦΑΡΜΟΓΗ ΤΟΥ VRPP ΣΤΟ ΧΩΡΟ ΤΗΣ ΝΑΥΤΙΛΙΑΣ

Το VRPP μπορεί να βρει εφαρμογή στο χώρο της ναυτιλίας. Εδώ θα ασχοληθούμε με ένα υποθετικό σενάριο το οποίο όμως είναι βασισμένο σε αρκετά πραγματικά στοιχεία.

Λόγω περιορισμένων βροχοπτώσεων τα νησιά του Αιγαίου αντιμετωπίζουν πρόβλημα λειψυδρίας. Η κυβέρνηση αποφασίζει ότι για να αντιμετωπιστεί το φαινόμενο, η ποσότητα του μεταφερόμενου νερού προς τα 15 νησιά με το μεγαλύτερο πρόβλημα λειψυδρίας πρέπει να αυξηθεί κατά 40%. Έτσι, κηρύττει διαγωνισμό στον οποίο οι εταιρείες που συμμετάσχουν δηλώνουν την προθυμία τους να μεταφέρουν πόσιμο νερό από το λιμάνι του Λαυρίου, Ελευσίνας η της Ρόδου σε ένα ή περισσότερα από τα νησιά. Η ποσότητα που πρέπει να μεταφέρουν είναι συγκεκριμένη και η μεταφορά πρέπει να γίνεται κάθε 2 εβδομάδες. Η κυβέρνηση είναι διατεθειμένη να πληρώσει 6,37 ευρώ για την μεταφορά κάθε μετρικού τόνου πόσιμου νερού

Εταιρεία διαθέτει 2 υδροφόρες που λειτουργούν στον χώρο του Αιγαίου με έδρα το Λαύριο. Το πρόγραμμα λειτουργίας των υδροφόρων επιτρέπει την συμμετοχή τους στον διαγωνισμό αφού έχουν αρκετό «ελεύθερο» χρόνο για ένα επιπλέον δρομολόγιο κάθε 2 εβδομάδες. Οι υδροφόρες έχουν μεταφορική ικανότητα 2704 τόνων η κάθε μία και καταναλώνουν πετρέλαιο που αντιστοιχεί σε περίπου 9,33 ευρώ ανά ναυτικό μίλι.

Η εταιρεία εξετάζει την συμμετοχή της στον διαγωνισμό. Επειδή οι υδροφόρες έχουν μικρότερη χωρητικότητα από την συνολική ζήτηση όλων των νησιών, αναζητά την επιλογή αυτών που θα μεγιστοποιήσουν τα κέρδη της.

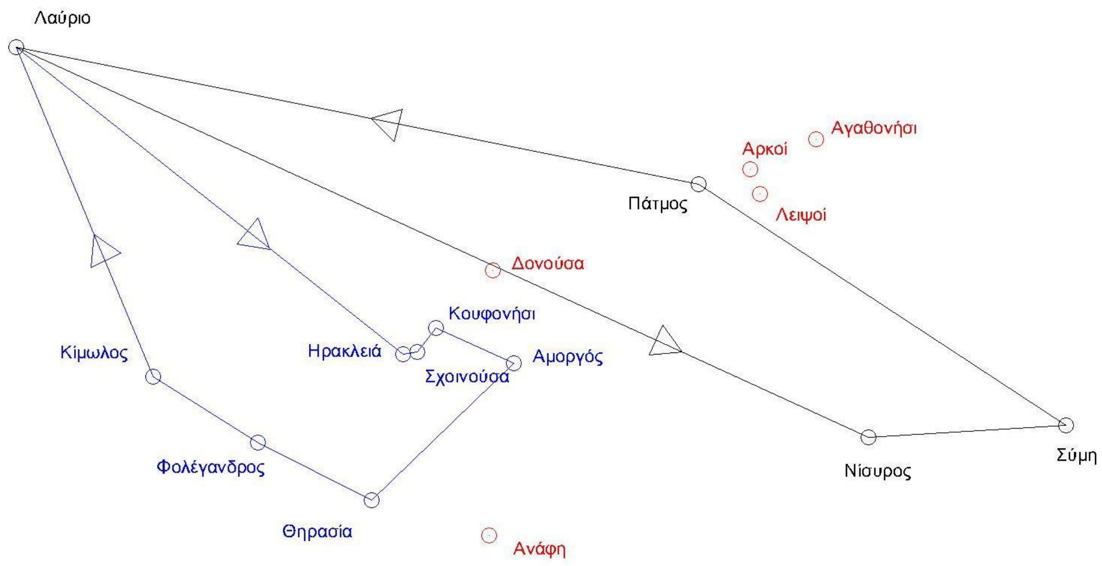
Στην περίπτωση αυτή θεωρούμε ότι τα πάγια κόστη χρήσης του πλοίου δεν εξαρτώνται από την διαδρομή που κάνει κάθε πλοίο. Οι μισθοί δηλαδή του πληρώματος, η ασφάλιση του πλοίου κτλ. δεν θα επηρεαστούν από ένα παραπάνω δρομολόγιο ανά 2 εβδομάδες. Το μόνο δηλαδή που θα προσμετρήσουμε ως κόστος είναι η κατανάλωση σε καύσιμα.

Η αναγκαία μεταφερόμενη ποσότητα πόσιμου νερού σε κάθε νησί δίνεται από τον παρακάτω πίνακα.

Λαύριο	0.0	Κουφονήσια	229.92
Πάτμος	793.94	Αγαθονήσι	103.93
Σύμη	1637.09	Θηρασία	168.36
Νίσυρος	251.16	Σχοινούσα	129.41
Ανάφη	171.50	Δονούσα	102.40
Λειψοί	438.48	Ηρακλειά	94.86
Κίμωλος	483.09	Αρκοί	33.92
Αμοργός	1167.83	**σύνολο**	**6214.90**
Φολέγανδρος	419.01		

πίνακας 11 αναγκαία μεταφερόμενη ποσότητα νερού της εφαρμογής στη ναυτιλία

Τα αποτελέσματα που πήραμε σε γραφική μορφή είναι:



σχήμα 9 αποτελέσματα εφαρμογής στη ναυτιλία

Όπως ήταν αναμενόμενο, τα πλοία δεν επισκέπτονται όλα τα νησιά, αφού δεν το επιτρέπει η χωρητικότητα τους. Τα πλοία σχεδόν εξαντλούν την χωρητικότητα τους (2672,2 με ολική 2704 και 2692 με ολική 2704) κάτι που σημαίνει ότι η τιμή των 6,37 ευρώ ανά μεταφερόμενο τόνο είναι ιδιαίτερα ελκυστική και η εταιρεία θα μπορούσε να κάνει προσφορά και σε μικρότερη τιμή. Βέβαια, δεν πρέπει να ξεχνάμε ότι στο κόστος δεν έχουν συμπεριληφθεί καθόλου τα πάγια κόστη κάτι που πρέπει να προσμετρηθεί αν γίνει καλύτερη προσφορά.

ΣΥΜΠΕΡΑΣΜΑΤΑ – ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΜΕΛΕΤΗ

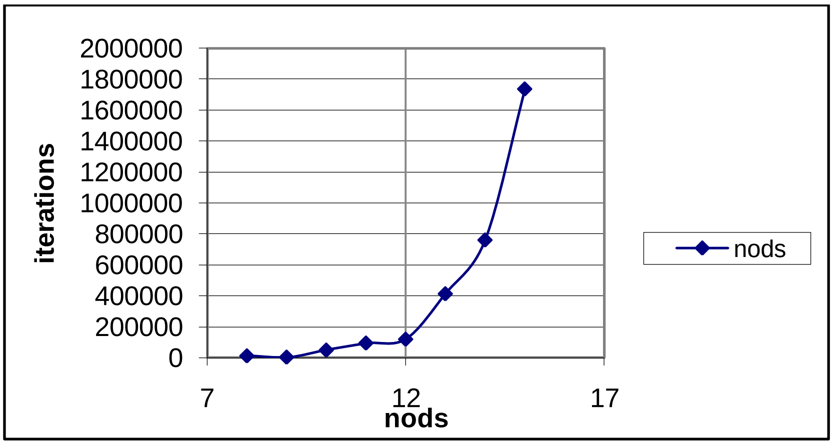
Ένα μεγάλο μέρος της μελέτης που έγινε για το VRPP αναλώθηκε στην ανάλυση ευαισθησίας. Από τα δοκιμαστικά τρεξίματα του προγράμματος είναι φανερό πως ο υπολογιστικός χρόνος που απαιτείται, εκτός από το μέγεθος του προβλήματος, εξαρτάται σε έναν σημαντικό βαθμό από τα δεδομένα που θα εισαχθούν στο πρόγραμμα. Η διαφορά, αυτή, σε υπολογιστικό χρόνο είναι αρκετά μεγάλη και δεν πρέπει να αγνοηθεί.

Μία λογική εξήγηση είναι ότι υπάρχουν ασύμμετρα δεδομένα που φωτογραφίζουν την λύση. Φανταστείτε, δηλαδή, ένα δίκτυο πόλεων με κεντρικό κόμβο την Αθήνα, κάποιες πόλεις στον χώρο της Ελλάδας και κάποιες άλλες στην άλλη πλευρά της γης. Είναι λογικό το πρόγραμμα, γρήγορα, να απορρίψει αυτές τις «άλλες» πόλεις και έτσι το πρόβλημα να μετασχηματιστεί σε άλλο μικρότερης τάξης που περιλαμβάνει μόνο τις ελληνικές πόλεις και να έχουμε γρήγορα αποτέλεσμα.

Βέβαια, υπάρχουν περιπτώσεις που η διαφορά σε υπολογιστικό χρόνο δεν εξηγείται τόσο εύκολα. Για δεδομένα φαινομενικά ίδια με ελάχιστες διαφορές είχαμε σημαντικές αποκλίσεις. Μία πιθανή εξήγηση είναι ότι το LINGO ξεκινάει από μία τυχαία λύση και αυτή παίζει σημαντικό ρόλο στην διαδικασία επίλυσης. Χαρακτηριστικό είναι ότι για το ίδιο ακριβώς πρόβλημα ο χρόνος που έκανε το LINGO για να το λύσει είχε μικρές αλλά αισθητές αποκλίσεις.

Αυτό που έγινε, επίσης, κατανοητό από την παρούσα μελέτη είναι ότι το VRPP είναι ένα πολύ δύσκολο πρόβλημα. Ακόμα και για προβλήματα σχετικά μικρού μεγέθους απαιτεί μεγάλη υπολογιστική δύναμη. Ένας σύγχρονος υπολογιστής για προβλήματα μεγέθους 15-18 κόμβων (μέγεθος για τα οποίο έγινε η παρούσα μελέτη) θα δώσει αποτελέσματα, συνήθως, μετά από αρκετές ώρες.

Είναι γνωστό από παλαιότερες μελέτες ότι το VRPP είναι τύπου NP-hard. Τα πειραματικά αποτελέσματα από την παρούσα μελέτη δεν θα μπορούσαν να ήταν διαφορετικά. Το ακόλουθο διάγραμμα, τα δεδομένα του οποίου είναι πειραματικά, δείχνει πως αυξάνεται ο υπολογιστικός χρόνος ανάλογα με τον αριθμό τον κόμβων, στην περίπτωση μας.



διάγραμμα 11 επαναλήψεις προς συνολικό αριθμό κόμβων

Την παραπάνω καμπύλη το excel την προσεγγίζει ως $y=6.004e^{0,842x}$, ο υπολογιστικός χρόνος δηλαδή αυξάνει εκθετικά όσο αυξάνει το μέγεθος του προβλήματος. Αυτό επιβεβαιώνει τον ορισμό του VRPP ως πρόβλημα τύπου NP-Hard.

Βάση των υπολογιστικών χρόνων που χρειάστηκαν ήταν εμφανές ότι το FS-VRPP είναι ένα πρόβλημα αρκετά πιο πολύπλοκο και σύνθετο από το VRPP. Κάτι αναμενόμενο αφού εκτός της επιλογής των βέλτιστων διαδρομών, εδώ, υπάρχει και η επιλογή του βέλτιστου στόλου. Έτσι, το FS-VRPP χρειάζεται πολλαπλάσιο υπολογιστικό χρόνο ακόμα και για προβλήματα πολύ μικρότερου μεγέθους σε σχέση με το VRPP.

Στο σημείο αυτό γίνεται κατανοητό ότι για μεγαλύτερα προβλήματα είναι θεμιτό να γίνει μία προσπάθεια μείωσης του υπολογιστικού χρόνου. Αυτό μπορεί να γίνει με δύο τρόπους. Ο πρώτος είναι με την χρησιμοποίηση ενός πιο σύγχρονου και δυνατού υπολογιστικού συστήματος. Βέβαια υπάρχουν όρια και ακόμα και το πιο σύγχρονο σύστημα της αγοράς θα μειώσει τον υπολογιστικό χρόνο, στην καλύτερη περίπτωση, στο 1/5. Ο δεύτερος τρόπος είναι να γίνει προσπάθεια εύρεσης αλγορίθμων που βελτιώνουν τον υπολογιστικό χρόνο (heuristics).

Τα τελευταία χρόνια, μάλιστα, η τάση που επικρατεί είναι η χρησιμοποίηση αλγορίθμων που βρίσκουν λύσεις αρκετά κοντά στην βέλτιστη με το πλεονέκτημα ότι την βρίσκουν αρκετά πιο γρήγορα. Ο συνδυασμός neighborhood search με tabu search, έτσι ώστε να αποκλείονται οι ανεπιθύμητοι κόμβοι που δεν συμβάλλουν στην αύξηση του κέρδους φαίνεται αρκετά ελκυστική ιδέα κάτι που επιβεβαιώνει και η δουλειά του Archetti.

Όσο αφορά την περίπτωση του FS-VRPP για προβλήματα μεγαλύτερου μεγέθους προτείνεται η επίλυση του με τον κλασικό τρόπο που γίνεται έως σήμερα. Ο διαχωρισμός του, δηλαδή, σε δύο είδη προβλημάτων: επιλογή βέλτιστου στόλου και δρομολόγηση οχημάτων. Βέβαια, επειδή σε αυτού του είδους τα προβλήματα δεν είναι γνωστός εκ των προτέρων ο αριθμός των συνολικών πελατών που θα εξυπηρετηθούν, συνεπώς και η ολική ζήτηση, αυτός ο τρόπος μπορεί να οδηγήσει σε λάθος επιλογή στόλου και σε αυτή την περίπτωση πρέπει να επανεκτιμηθεί.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Archetti, C., A. Hertz, and M. G. Speranza (2007). Metaheurisitics for the team orienteering problem. Journal of Heuristics 13, 49–76.

Feillet D, Dejax P and Gendreau M (2005) Traveling salesman problems with profits. Transportation Science 39:188−205.

G. Gutin, A.P. Punnen (2002), The Traveling Salesman Problem and its Variations, Kluwer Academic, Norwell, MA

Tang H and Miller-Hooks E (2005) A tabu search heuristic for the team orienteering problem. Computers & Operations Research 32: 1379–1407.

The VRP WEB (2007) - http://neo.lcc.uma.es/radi-aeb/WebVRP/ - Accessed September 16th, 2007

Toth, P. and D. Vigo (2002). The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications, SIAM Publishing.

Winston, W. (2004) Operations Research: Applications and Algorithms.*Brooks/Cole – Thompson Learning*.

ΠΑΡΑΡΤΗΜΑ

ΔΕΔΟΜΕΝΑ ΠΑΡΑΔΕΙΓΜΑΤΩΝ

Δεδομένα για την ανάλυση ευαισθησίας

Επειδή έγιναν πολυάριθμα δοκιμαστικά τρεξίματα για την ανάλυση ευαισθησίας δεν θα τα παραθέσουμε όλα, αφού κάτι τέτοιο θα ήταν κουραστικό για τον αναγνώστη. Αντ' αυτού θα δώσουμε τα δεδομένα που είναι κοινά για όλα τα τρεξίματα.

Στα περισσότερα παραδείγματα χρησιμοποιήθηκαν 3 οχήματα συνολικά χωρητικότητας 330 πακέτων (το καθένα). Όπου αναφέρεται άλλη τιμή από αυτήν, φυσικά ισχύει η τιμή που αναφέρεται.

Για τις πόλεις ισχύουν:

	ζήτηση σε πακέτα
Madrid	0
Barcelona	328
Valencia	103
Sevilla	91
Málaga	72
Bilbao	67
Gijón	60
Alicante	47
Zaragoza	46
Murcia	40
Granada	32
Vigo	30
Cadiz	29
San Sebastian	28
A Corunna	28
σύνολο	**1001**

Κεντρικός κόμβος : Madrid

Οι αποδοχές σε κάθε πόλη υπολογίζονται από το γινόμενο ζήτηση σε πακέτο * αποδοχές ανά πακέτο

κόστη διαδρομών

	Madrid	Barcelona	Valencia	Sevilla	Málaga	Bilbao	Gijón	Alicante	Zaragoza	Murcia	Granada	Vigo	Cadiz	San Sebastian	A Corunna	Valladolid	Tarragona	Córdoba
Madrid	5000	73.67	69.5	62.55	61.16	44.48	55.59	48.65	37.52	45.87	48.65	66.72	75.5	55.59	66.72	23.63	63.94	45.87
Barcelona	73.67	5000	41.7	132.4	118.15	72.28	102.86	63.94	36.84	73.67	104.24	136.22	137.6	66.72	129.26	87.57	11.12	102.86
Valencia	69.5	41.7	5000	88.96	75.5	76.44	97.3	20.85	37.52	30.58	61.16	109.8	94.52	93.13	109.8	66.72	30.58	59.77
Sevilla	62.55	132.4	88.96	5000	23.63	98.69	95.91	68.11	95.91	95.77	29.19	107.2	13.89	108.41	125.1	65.33	120.93	15.29
Málaga	61.16	118.15	75.5	23.63	5000	107.2	116.76	52.81	94.52	44.48	13.89	132.4	29.19	118.15	127.88	84.79	107.2	18.7
Bilbao	44.48	72.28	76.44	98.69	107.2	5000	30.58	94.52	36.14	91.74	94.52	80.61	112.58	11.12	73.67	31.97	62.55	91.74
Gijón	55.59	102.86	97.3	95.91	116.76	30.58	5000	104.24	66.72	102.86	104.2	47.26	109.8	41.7	31.97	37.52	93.13	100.8
Alicante	48.65	63.94	20.85	68.11	52.81	94.52	104.24	5000	58.38	9.72	41.7	116.76	72.28	93.13	116.76	73.67	52.81	56.98
Zaragoza	37.52	36.84	37.52	95.91	94.52	36.14	66.72	58.38	5000	68.11	82.1	100.8	109.8	30.58	93.13	51.43	27.79	79.22
Murcia	45.87	73.67	30.58	95.77	44.48	91.74	102.86	9.72	68.11	5000	31.97	113.97	63.4	102.86	115.36	70.89	62.55	48.65
Granada	48.65	104.24	61.16	29.19	13.89	94.52	104.2	41.7	82.1	31.97	5000	116.76	33.36	105.63	116.76	73.67	93.13	18.7
Vigo	66.72	136.22	109.8	107.2	132.4	80.61	47.26	116.76	100.8	113.97	116.76	5000	122.32	90.35	18.7	48.65	126.49	113.97
Cadiz	75.5	137.6	94.52	13.89	29.19	112.58	109.8	72.28	109.8	63.4	33.36	122.32	5000	123.71	139	80.61	125.1	29.19
San Sebastian	55.59	66.72	93.13	108.41	118.15	11.12	41.7	93.13	30.58	102.86	105.63	90.35	123.71	5000	84.79	43.9	56.98	102.86
A Corunna	66.72	129.26	109.8	125.1	127.88	73.67	31.97	116.76	93.13	115.36	116.76	18.7	139	84.79	5000	50.4	119.54	112.58
Valladolid	23.63	87.57	66.72	65.33	84.79	31.97	37.52	73.67	51.43	70.89	73.67	48.65	80.61	43.9	50.4	5000	77.83	69.5
Tarragona	63.94	11.12	30.58	120.93	107.2	62.55	93.13	52.81	27.79	62.55	93.13	126.49	125.1	56.98	119.54	77.83	5000	91.74
Córdoba	45.87	102.86	59.77	15.29	18.7	91.74	100.8	56.98	79.22	48.65	18.7	113.97	29.19	102.86	112.58	69.5	91.74	5000

1ο παράδειγμα για το PVRPP

3 οχήματα συνολικά χωρητικότητας 450 πακέτων (το καθένα).

Για τις πόλεις ισχύουν:

	ζήτηση σε πακέτα	αποδοχές
Αθήνα	0	0
Θεσσαλονίκη	444	355.2
Πάτρα	103	82.4
Λάρισα	77	61.6
Βόλος	68	54.4
Ιωάννινα	43	34.4
Χαλκίδα	36	28.8
Καβάλα	31	24.8
Καλαμάτα	31	24.8
Σέρρες	30	24
Κατερίνη	29	23.2
Τρίκαλα	29	23.2
Λαμία	27	21.6
Αλεξανδρούπολη	26	20.8
Ξάνθη	25	20
σύνολο	**999**	**799.2**

Κεντρικός κόμβος : Αθήνα

κόστη διαδρομών

	Αθήνα	Θεσσαλονίκη	Πάτρα	Λάρισα	Βόλος	Ιωάννινα	Χαλκίδα	Καβάλα	Καλαμάτα	Σέρρες	Κατερίνη	Τρίκαλα	Λαμία	Αλεξανδρούπολη	Ξάνθη
Αθήνα	460	47.38	19.872	33.028	29.808	40.296	8.096	62.56	26.128	56.304	41.032	30.452	19.688	78.568	67.712
Θεσσαλονίκη	47.38	460	42.228	17.112	19.872	34.04	41.952	15.18	65.78	7.82	9.292	20.24	30.544	31.832	20.332
Πάτρα	19.872	42.228	460	31.096	28.796	22.54	26.036	57.408	19.78	53.452	39.192	28.888	18.032	76.176	65.964
Λάρισα	33.028	17.112	31.096	460	5.52	19.504	27.232	32.292	51.796	22.356	8.096	5.704	13.616	45.356	34.868
Βόλος	29.808	19.872	28.796	5.52	460	25.208	24.472	35.236	48.392	27.784	13.708	11.132	10.58	51.152	40.204
Ιωάννινα	40.296	34.04	22.54	19.504	25.208	460	38.916	49.22	42.044	39.928	27.324	13.8	24.564	64.584	52.44
Χαλκίδα	8.096	41.952	26.036	27.232	24.472	38.916	460	57.132	27.048	4.968	35.788	25.3	14.444	73.784	62.284
Καβάλα	62.56	15.18	57.408	32.292	35.236	49.22	57.132	460	80.776	9.752	24.472	35.42	45.724	16.284	5.152
Καλαμάτα	26.128	65.78	19.78	51.796	48.392	42.044	27.048	80.776	460	73.14	59.432	48.852	38.18	97.06	85.836
Σέρρες	56.304	7.82	53.452	22.356	27.784	39.928	4.968	9.752	73.14	460	14.26	28.06	35.42	24.932	14.26
Κατερίνη	41.032	9.292	39.192	8.096	13.708	27.324	35.788	24.472	59.432	14.26	460	13.984	21.16	37.812	26.772
Τρίκαλα	30.452	20.24	28.888	5.704	11.132	13.8	25.3	35.42	48.852	28.06	13.984	460	10.856	50.968	40.572
Λαμία	19.688	30.544	18.032	13.616	10.58	24.564	14.444	45.724	38.18	35.42	21.16	10.856	460	59.156	47.932
Αλεξανδρούπολη	78.568	31.832	76.176	45.356	51.152	64.584	73.784	16.284	97.06	24.932	37.812	50.968	59.156	460	11.224
Ξάνθη	67.712	20.332	65.964	34.868	40.204	52.44	62.284	5.152	85.836	14.26	26.772	40.572	47.932	11.224	460

2ο παράδειγμα για το PVRPP

3 οχήματα συνολικά χωρητικότητας 450 πακέτων (το καθένα).

Για τις πόλεις ισχύουν:

	ζήτηση σε πακέτα	αποδοχές
Αθήνα	0	0
Θεσσαλονίκη	444	577.2
Πάτρα	103	133.9
Λάρισα	77	100.1
Βόλος	68	88.4
Ιωάννινα	43	55.9
Χαλκίδα	36	46.8
Καβάλα	31	40.3
Καλαμάτα	31	40.3
Σέρρες	30	39
Κατερίνη	29	37.7
Τρίκαλα	29	37.7
Λαμία	27	35.1
Αλεξανδρούπολη	26	33.8
Ξάνθη	25	32.5
σύνολο	**999**	**1298.7**

Κεντρικός κόμβος : Αθήνα

κόστη διαδρομών

	Αθήνα	Θεσσαλονίκη	Πάτρα	Λάρισα	Βόλος	Ιωάννινα	Χαλκίδα	Καβάλα	Καλαμάτα	Σέρρες	Κατερίνη	Τρίκαλα	Λαμία	Αλεξανδρούπολη	Ξάνθη
Αθήνα	460	47.38	19.872	33.028	29.808	40.296	8.096	62.56	26.128	56.304	41.032	30.452	19.688	78.568	67.712
Θεσσαλονίκη	47.38	460	42.228	17.112	19.872	34.04	41.952	15.18	65.78	7.82	9.292	20.24	30.544	31.832	20.332
Πάτρα	19.872	42.228	460	31.096	28.796	22.54	26.036	57.408	19.78	53.452	39.192	28.888	18.032	76.176	65.964
Λάρισα	33.028	17.112	31.096	460	5.52	19.504	27.232	32.292	51.796	22.356	8.096	5.704	13.616	45.356	34.868
Βόλος	29.808	19.872	28.796	5.52	460	25.208	24.472	35.236	48.392	27.784	13.708	11.132	10.58	51.152	40.204
Ιωάννινα	40.296	34.04	22.54	19.504	25.208	460	38.916	49.22	42.044	39.928	27.324	13.8	24.564	64.584	52.44
Χαλκίδα	8.096	41.952	26.036	27.232	24.472	38.916	460	57.132	27.048	4.968	35.788	25.3	14.444	73.784	62.284
Καβάλα	62.56	15.18	57.408	32.292	35.236	49.22	57.132	460	80.776	9.752	24.472	35.42	45.724	16.284	5.152
Καλαμάτα	26.128	65.78	19.78	51.796	48.392	42.044	27.048	80.776	460	73.14	59.432	48.852	38.18	97.06	85.836
Σέρρες	56.304	7.82	53.452	22.356	27.784	39.928	4.968	9.752	73.14	460	14.26	28.06	35.42	24.932	14.26
Κατερίνη	41.032	9.292	39.192	8.096	13.708	27.324	35.788	24.472	59.432	14.26	460	13.984	21.16	37.812	26.772
Τρίκαλα	30.452	20.24	28.888	5.704	11.132	13.8	25.3	35.42	48.852	28.06	13.984	460	10.856	50.968	40.572
Λαμία	19.688	30.544	18.032	13.616	10.58	24.564	14.444	45.724	38.18	35.42	21.16	10.856	460	59.156	47.932
Αλεξανδρούπολη	78.568	31.832	76.176	45.356	51.152	64.584	73.784	16.284	97.06	24.932	37.812	50.968	59.156	460	11.224
Ξάνθη	67.712	20.332	65.964	34.868	40.204	52.44	62.284	5.152	85.836	14.26	26.772	40.572	47.932	11.224	460

3ο παράδειγμα για το PVRPP

3 οχήματα συνολικά χωρητικότητας 500 πακέτων (το καθένα).

Για τις πόλεις ισχύουν:

	ζήτηση σε πακέτα	αποδοχές
Αθήνα	0	0
Θεσσαλονίκη	444	577.2
Πάτρα	103	133.9
Λάρισα	77	100.1
Βόλος	68	88.4
Ιωάννινα	43	55.9
Χαλκίδα	36	46.8
Καβάλα	31	40.3
Καλαμάτα	31	40.3
Σέρρες	30	39
Κατερίνη	29	37.7
Τρίκαλα	29	37.7
Λαμία	27	35.1
Αλεξανδρούπολη	26	33.8
Ξάνθη	25	32.5
σύνολο	**999**	**1298.7**

Κεντρικός κόμβος : Αθήνα

κόστη διαδρομών

	Αθήνα	Θεσσαλονίκη	Πάτρα	Λάρισα	Βόλος	Ιωάννινα	Χαλκίδα	Καβάλα	Καλαμάτα	Σέρρες	Κατερίνη	Τρίκαλα	Λαμία	Αλεξανδρούπολη	Ξάνθη
Αθήνα	460	47.38	19.872	33.028	29.808	40.296	8.096	62.56	26.128	56.304	41.032	30.452	19.688	78.568	67.712
Θεσσαλονίκη	47.38	460	42.228	17.112	19.872	34.04	41.952	15.18	65.78	7.82	9.292	20.24	30.544	31.832	20.332
Πάτρα	19.872	42.228	460	31.096	28.796	22.54	26.036	57.408	19.78	53.452	39.192	28.888	18.032	76.176	65.964
Λάρισα	33.028	17.112	31.096	460	5.52	19.504	27.232	32.292	51.796	22.356	8.096	5.704	13.616	45.356	34.868
Βόλος	29.808	19.872	28.796	5.52	460	25.208	24.472	35.236	48.392	27.784	13.708	11.132	10.58	51.152	40.204
Ιωάννινα	40.296	34.04	22.54	19.504	25.208	460	38.916	49.22	42.044	39.928	27.324	13.8	24.564	64.584	52.44
Χαλκίδα	8.096	41.952	26.036	27.232	24.472	38.916	460	57.132	27.048	4.968	35.788	25.3	14.444	73.784	62.284
Καβάλα	62.56	15.18	57.408	32.292	35.236	49.22	57.132	460	80.776	9.752	24.472	35.42	45.724	16.284	5.152
Καλαμάτα	26.128	65.78	19.78	51.796	48.392	42.044	27.048	80.776	460	73.14	59.432	48.852	38.18	97.06	85.836
Σέρρες	56.304	7.82	53.452	22.356	27.784	39.928	4.968	9.752	73.14	460	14.26	28.06	35.42	24.932	14.26
Κατερίνη	41.032	9.292	39.192	8.096	13.708	27.324	35.788	24.472	59.432	14.26	460	13.984	21.16	37.812	26.772
Τρίκαλα	30.452	20.24	28.888	5.704	11.132	13.8	25.3	35.42	48.852	28.06	13.984	460	10.856	50.968	40.572
Λαμία	19.688	30.544	18.032	13.616	10.58	24.564	14.444	45.724	38.18	35.42	21.16	10.856	460	59.156	47.932
Αλεξανδρούπολη	78.568	31.832	76.176	45.356	51.152	64.584	73.784	16.284	97.06	24.932	37.812	50.968	59.156	460	11.224
Ξάνθη	67.712	20.332	65.964	34.868	40.204	52.44	62.284	5.152	85.836	14.26	26.772	40.572	47.932	11.224	460

παράδειγμα για το TOP

3 οχήματα συνολικά χωρητικότητας 450 πακέτων (το καθένα).

Για τις πόλεις ισχύουν:

	ζήτηση σε πακέτα	αποδοχές
Αθήνα	0	0
Θεσσαλονίκη	444	355.2
Πάτρα	103	82.4
Λάρισα	77	61.6
Βόλος	68	54.4
Ιωάννινα	43	34.4
Χαλκίδα	36	28.8
Καβάλα	31	24.8
Καλαμάτα	31	24.8
Σέρρες	30	24
Κατερίνη	29	23.2
Τρίκαλα	29	23.2
Λαμία	27	21.6
Αλεξανδρούπολη	26	20.8
Ξάνθη	25	20
σύνολο	**999**	**799.2**

Κεντρικός κόμβος : Αθήνα

Μέγιστο κόστος : 321,86

κόστη διαδρομών

	Αθήνα	Θεσσαλονίκη	Πάτρα	Λάρισα	Βόλος	Ιωάννινα	Χαλκίδα	Καβάλα	Καλαμάτα	Σέρρες	Κατερίνη	Τρίκαλα	Λαμία	Αλεξανδρούπολη	Ξάνθη
Αθήνα	460	47.38	19.872	33.028	29.808	40.296	8.096	62.56	26.128	56.304	41.032	30.452	19.688	78.568	67.712
Θεσσαλονίκη	47.38	460	42.228	17.112	19.872	34.04	41.952	15.18	65.78	7.82	9.292	20.24	30.544	31.832	20.332
Πάτρα	19.872	42.228	460	31.096	28.796	22.54	26.036	57.408	19.78	53.452	39.192	28.888	18.032	76.176	65.964
Λάρισα	33.028	17.112	31.096	460	5.52	19.504	27.232	32.292	51.796	22.356	8.096	5.704	13.616	45.356	34.868
Βόλος	29.808	19.872	28.796	5.52	460	25.208	24.472	35.236	48.392	27.784	13.708	11.132	10.58	51.152	40.204
Ιωάννινα	40.296	34.04	22.54	19.504	25.208	460	38.916	49.22	42.044	39.928	27.324	13.8	24.564	64.584	52.44
Χαλκίδα	8.096	41.952	26.036	27.232	24.472	38.916	460	57.132	27.048	4.968	35.788	25.3	14.444	73.784	62.284
Καβάλα	62.56	15.18	57.408	32.292	35.236	49.22	57.132	460	80.776	9.752	24.472	35.42	45.724	16.284	5.152
Καλαμάτα	26.128	65.78	19.78	51.796	48.392	42.044	27.048	80.776	460	73.14	59.432	48.852	38.18	97.06	85.836
Σέρρες	56.304	7.82	53.452	22.356	27.784	39.928	4.968	9.752	73.14	460	14.26	28.06	35.42	24.932	14.26
Κατερίνη	41.032	9.292	39.192	8.096	13.708	27.324	35.788	24.472	59.432	14.26	460	13.984	21.16	37.812	26.772
Τρίκαλα	30.452	20.24	28.888	5.704	11.132	13.8	25.3	35.42	48.852	28.06	13.984	460	10.856	50.968	40.572
Λαμία	19.688	30.544	18.032	13.616	10.58	24.564	14.444	45.724	38.18	35.42	21.16	10.856	460	59.156	47.932
Αλεξανδρούπολη	78.568	31.832	76.176	45.356	51.152	64.584	73.784	16.284	97.06	24.932	37.812	50.968	59.156	460	11.224
Ξάνθη	67.712	20.332	65.964	34.868	40.204	52.44	62.284	5.152	85.836	14.26	26.772	40.572	47.932	11.224	460

παράδειγμα για το PC-VRPP

3 οχήματα συνολικά χωρητικότητας 450 πακέτων (το καθένα).

Για τις πόλεις ισχύουν:

	ζήτηση σε πακέτα	αποδοχές
Αθήνα	0	0
Θεσσαλονίκη	444	355.2
Πάτρα	103	82.4
Λάρισα	77	61.6
Βόλος	68	54.4
Ιωάννινα	43	34.4
Χαλκίδα	36	28.8
Καβάλα	31	24.8
Καλαμάτα	31	24.8
Σέρρες	30	24
Κατερίνη	29	23.2
Τρίκαλα	29	23.2
Λαμία	27	21.6
Αλεξανδρούπολη	26	20.8
Ξάνθη	25	20
σύνολο	**999**	**799.2**

Κεντρικός κόμβος : Αθήνα

ελάχιστα έσοδα : 703,6

κόστη διαδρομών

	Αθήνα	Θεσσαλονίκη	Πάτρα	Λάρισα	Βόλος	Ιωάννινα	Χαλκίδα	Καβάλα	Καλαμάτα	Σέρρες	Κατερίνη	Τρίκαλα	Λαμία	Αλεξανδρούπολη	Ξάνθη
Αθήνα	460	47.38	19.872	33.028	29.808	40.296	8.096	62.56	26.128	56.304	41.032	30.452	19.688	78.568	67.712
Θεσσαλονίκη	47.38	460	42.228	17.112	19.872	34.04	41.952	15.18	65.78	7.82	9.292	20.24	30.544	31.832	20.332
Πάτρα	19.872	42.228	460	31.096	28.796	22.54	26.036	57.408	19.78	53.452	39.192	28.888	18.032	76.176	65.964
Λάρισα	33.028	17.112	31.096	460	5.52	19.504	27.232	32.292	51.796	22.356	8.096	5.704	13.616	45.356	34.868
Βόλος	29.808	19.872	28.796	5.52	460	25.208	24.472	35.236	48.392	27.784	13.708	11.132	10.58	51.152	40.204
Ιωάννινα	40.296	34.04	22.54	19.504	25.208	460	38.916	49.22	42.044	39.928	27.324	13.8	24.564	64.584	52.44
Χαλκίδα	8.096	41.952	26.036	27.232	24.472	38.916	460	57.132	27.048	4.968	35.788	25.3	14.444	73.784	62.284
Καβάλα	62.56	15.18	57.408	32.292	35.236	49.22	57.132	460	80.776	9.752	24.472	35.42	45.724	16.284	5.152
Καλαμάτα	26.128	65.78	19.78	51.796	48.392	42.044	27.048	80.776	460	73.14	59.432	48.852	38.18	97.06	85.836
Σέρρες	56.304	7.82	53.452	22.356	27.784	39.928	4.968	9.752	73.14	460	14.26	28.06	35.42	24.932	14.26
Κατερίνη	41.032	9.292	39.192	8.096	13.708	27.324	35.788	24.472	59.432	14.26	460	13.984	21.16	37.812	26.772
Τρίκαλα	30.452	20.24	28.888	5.704	11.132	13.8	25.3	35.42	48.852	28.06	13.984	460	10.856	50.968	40.572
Λαμία	19.688	30.544	18.032	13.616	10.58	24.564	14.444	45.724	38.18	35.42	21.16	10.856	460	59.156	47.932
Αλεξανδρούπολη	78.568	31.832	76.176	45.356	51.152	64.584	73.784	16.284	97.06	24.932	37.812	50.968	59.156	460	11.224
Ξάνθη	67.712	20.332	65.964	34.868	40.204	52.44	62.284	5.152	85.836	14.26	26.772	40.572	47.932	11.224	460

παράδειγμα για το FS-VRPP

για τα οχήματα ισχύει:

Όχημα	αρχικό κόστος	κατανάλωση	χωρητικότητα
1	138.89	10.23l/km	315
2	153.75	9.4l/km	170
3	156.80	11.76l/km	675

για τις πόλεις ισχύουν:

	ζήτηση	αποδοχές
Θεσσαλονίκη	0	0
Πάτρα	206	164.8
Λάρισα	155	124
Βόλος	136	108.8
Ιωάννινα	86	68.8
Χαλκίδα	73	58.4
Καβάλα	63	50.4
Σέρρες	59	47.2
Κατερίνη	59	47.2
Τρίκαλα	58	46.4
Λαμία	54	43.2
Αλεξανδρούπολη	53	42.4
σύνολο	**1002**	**801.6**

Κεντρικός κόμβος : Θεσσαλονίκη

κόστη διαδρομών

	Θεσσαλονίκη			Πάτρα			Λάρισα			Βόλος			Ιωάννινα			Χαλκίδα		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Θεσσαλονίκη	511.5	470	588	46.9557	43.146	53.9784	19.0278	17.484	21.8736	22.0968	20.304	25.4016	37.851	34.78	43.512	46.6488	42.864	53.6256
Πάτρα	46.9557	43.146	53.9784	511.5	470	588	34.5774	31.772	39.7488	32.0199	29.422	36.8088	25.0635	23.03	28.812	28.9509	26.602	33.2808
Λάρισα	19.0278	17.484	21.8736	34.5774	31.772	39.7488	511.5	470	588	6.138	5.64	7.056	21.6876	19.928	24.9312	30.2808	27.824	34.8096
Βόλος	22.0968	20.304	25.4016	32.0199	29.422	36.8088	6.138	5.64	7.056	511.5	470	588	28.0302	25.756	32.2224	27.2118	25.004	31.2816
Ιωάννινα	37.851	34.78	43.512	25.0635	23.03	28.812	21.6876	19.928	24.9312	28.0302	25.756	32.2224	511.5	470	588	43.2729	39.762	49.7448
Χαλκίδα	46.6488	42.864	53.6256	28.9509	26.602	33.2808	30.2808	27.824	34.8096	27.2118	25.004	31.2816	43.2729	39.762	49.7448	511.5	470	588
Καβάλα	16.8795	15.51	19.404	63.8352	58.656	73.3824	35.9073	32.994	41.2776	39.1809	36.002	45.0408	54.7305	50.29	62.916	63.5283	58.374	73.0296
Σέρρες	8.6955	7.99	9.996	59.4363	54.614	68.3256	24.8589	22.842	28.5768	30.8946	28.388	35.5152	44.3982	40.796	51.0384	5.5242	5.076	6.3504
Κατερίνη	10.3323	9.494	11.8776	43.5798	40.044	50.0976	9.0024	8.272	10.3488	15.2427	14.006	17.5224	30.3831	27.918	34.9272	39.7947	36.566	45.7464
Τρίκαλα	22.506	20.68	25.872	32.1222	29.516	36.9264	6.3426	5.828	7.2912	12.3783	11.374	14.2296	15.345	14.1	17.64	28.1325	25.85	32.34
Λαμία	33.9636	31.208	39.0432	20.0508	18.424	23.0496	15.1404	13.912	17.4048	11.7645	10.81	13.524	27.3141	25.098	31.3992	16.0611	14.758	18.4632
Αλεξανδρούπολη	35.3958	32.524	40.6896	84.7044	77.832	97.3728	50.4339	46.342	57.9768	56.8788	52.264	65.3856	71.8146	65.988	82.5552	82.0446	75.388	94.3152

	Καβάλα			Σέρρες			Κατερίνη			Τρίκαλα			Λαμία			Αλεξανδρούπολη		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Θεσσαλονίκη	16.8795	15.51	19.404	8.6955	7.99	9.996	10.3323	9.494	11.8776	22.506	20.68	25.872	33.9636	31.208	39.0432	35.3958	32.524	40.6896
Πάτρα	63.8352	58.656	73.3824	59.4363	54.614	68.3256	43.5798	40.044	50.0976	32.1222	29.516	36.9264	20.0508	18.424	23.0496	84.7044	77.832	97.3728
Λάρισα	35.9073	32.994	41.2776	24.8589	22.842	28.5768	9.0024	8.272	10.3488	6.3426	5.828	7.2912	15.1404	13.912	17.4048	50.4339	46.342	57.9768
Βόλος	39.1809	36.002	45.0408	30.8946	28.388	35.5152	15.2427	14.006	17.5224	12.3783	11.374	14.2296	11.7645	10.81	13.524	56.8788	52.264	65.3856
Ιωάννινα	54.7305	50.29	62.916	44.3982	40.796	51.0384	30.3831	27.918	34.9272	15.345	14.1	17.64	27.3141	25.098	31.3992	71.8146	65.988	82.5552
Χαλκίδα	63.5283	58.374	73.0296	5.5242	5.076	6.3504	39.7947	36.566	45.7464	28.1325	25.85	32.34	16.0611	14.758	18.4632	82.0446	75.388	94.3152
Καβάλα	511.5	470	588	10.8438	9.964	12.4656	27.2118	25.004	31.2816	39.3855	36.19	45.276	50.8431	46.718	58.4472	18.1071	16.638	20.8152
Σέρρες	10.8438	9.964	12.4656	511.5	470	588	15.8565	14.57	18.228	31.2015	28.67	35.868	39.3855	36.19	45.276	27.7233	25.474	31.8696
Κατερίνη	27.2118	25.004	31.2816	15.8565	14.57	18.228	511.5	470	588	15.5496	14.288	17.8752	23.529	21.62	27.048	42.0453	38.634	48.3336
Τρίκαλα	39.3855	36.19	45.276	31.2015	28.67	35.868	15.5496	14.288	17.8752	511.5	470	588	12.0714	11.092	13.8768	56.6742	52.076	65.1504
Λαμία	50.8431	46.718	58.4472	39.3855	36.19	45.276	23.529	21.62	27.048	12.0714	11.092	13.8768	511.5	470	588	65.7789	60.442	75.6168
Αλεξανδρούπολη	18.1071	16.638	20.8152	27.7233	25.474	31.8696	42.0453	38.634	48.3336	56.6742	52.076	65.1504	65.7789	60.442	75.6168	511.5	470	588

η στήλη 1 κάθε πόλης αντιστοιχεί στο όχημα 1 κ.ο.κ.

Εφαρμογή στη ναυτιλία

2 υδροφόρα πλοία χωρητικότητας 2704 μετρικών τόνων (το καθένα)

για τα νησιά (κόμβους) ισχύουν:

	ζήτηση	αποδοχές
Λαύριο	0	0
Πάτμος	783.93	4993.68
Σύμη	16937.09	10428.29
Νίσυρος	251.16	1599.92
Ανάφη	171.5	1092.45
Λειψοί	438.48	2791.15
Κίμωλος	483.09	3077.27
Αμοργός	1167.83	7439.06
Φολέγανδρος	419.01	2669.10
Κουφονήσια	229.92	1464.60
Αγαθονήσι	103.93	662.03
Θηρασία	168.36	1072.44
Σχοινούσα	129.41	824.34
Δονούσα	102.40	652.27
Ηρακλειά	94.86	604.25
Αρκοί	33.92	216.09
σύνολο	6214.90	39588.92

Κεντρικός κόμβος : Λαύριο

κόστη διαδρομών

	Λαύριο	Πάτμος	Σύμη	Νίσυρος	Ανάφη	Λειψοί	Κίμωλος	Αμοργός	Φολέγανδρος	Κουφονήσια	Αγαθονήσι	Θηρασία	Σχοινούσα	Δονούσα	Ηρακλειά	Αρκοί
Λαύριο	9330000000	1532.7324	2414.0442	1942.4127	1395.8613	1561.0023	687.9009	1248.4473	904.3569	1146.7503	1641.8001	1183.2306	1126.2243	1111.0164	1097.208	1525.0818
Πάτμος	1532.7324	9330000000	1032.831	606.1701	931.3206	118.6776	1177.9125	563.1588	1094.1291	620.8182	247.1517	923.67	690.3267	477.3228	719.7162	117.4647
Σύμη	2414.0442	1032.831	9330000000	426.4743	1165.2237	976.2912	1770.6474	1202.1705	1616.7024	1297.803	1210.7541	1425.0642	1313.7573	1394.6484	1325.9796	1085.2656
Νίσυρος	1942.4127	606.1701	426.4743	9330000000	735.204	543.1926	1358.448	855.6543	1141.4322	913.9668	728.3931	986.181	883.7376	769.2585	898.3857	620.2584
Ανάφη	1395.8613	931.3206	1165.2237	735.204	9330000000	846.231	734.7375	478.9089	481.8012	473.7774	984.6882	273.4623	396.8982	550.3767	405.3885	862.092
Λειψοί	1561.0023	118.6776	976.2912	543.1926	846.231	9330000000	1234.1724	590.1225	1061.754	661.7769	192.198	952.6863	739.2159	533.3961	736.4169	100.2975
Κίμωλος	687.9009	1177.9125	1770.6474	1358.448	734.7375	1234.1724	9330000000	667.4682	238.5681	535.6353	1348.7448	504.1932	505.2195	679.4106	514.083	1222.0434
Αμοργός	1248.4473	563.1588	1202.1705	855.6543	478.9089	590.1225	667.4682	9330000000	515.8557	162.1554	769.0719	396.8049	203.7672	190.0521	216.456	622.8708
Φολέγανδρος	904.3569	1094.1291	1616.7024	1141.4322	481.8012	1061.754	238.5681	515.8557	9330000000	399.6039	1198.1586	254.1492	359.8581	544.0323	342.8775	1067.9118
Κουφονήσια	1146.7503	620.8182	1297.803	913.9668	473.7774	661.7769	535.6353	162.1554	399.6039	9330000000	816.0018	364.6164	84.7164	163.275	81.6375	688.9272
Αγαθονήσι	1641.8001	247.1517	1210.7541	728.3931	984.6882	192.198	1348.7448	769.0719	1198.1586	816.0018	9330000000	1138.9131	917.4189	705.4413	908.2755	186.5067
Θηρασία	1183.2306	923.67	1425.0642	986.181	273.4623	952.6863	504.1932	396.8049	254.1492	364.6164	1138.9131	9330000000	294.5481	503.1669	303.5049	977.9706
Σχοινούσα	1126.2243	690.3267	1313.7573	883.7376	396.8982	739.2159	505.2195	203.7672	359.8581	84.7164	917.4189	294.5481	9330000000	239.8743	29.6694	764.9667
Δονούσα	1111.0164	477.3228	1394.6484	769.2585	550.3767	533.3961	679.4106	190.0521	544.0323	163.275	705.4413	503.1669	239.8743	9330000000	227.0922	554.9484
Ηρακλειά	1097.208	719.7162	1325.9796	898.3857	405.3885	736.4169	514.083	216.456	342.8775	81.6375	908.2755	303.5049	29.6694	227.0922	9330000000	772.8972
Αρκοί	1525.0818	117.4647	1085.2656	620.2584	862.092	100.2975	1222.0434	622.8708	1067.9118	688.9272	186.5067	977.9706	764.9667	554.9484	772.8972	9330000000

Ο κώδικας του LINGO

The Profitable Vehicle Routing Problem with Profits

```
!code for the PVRPP made by Vasilis Tourloupis;

SETS:

CITY/1..18/:P,W,A;
SHIP/1..3/:B;
LINK(CITY,CITY,SHIP):X;
LINKZ(CITY,SHIP):Y,U;
LINKS(CITY,CITY):DIST;
ENDSETS

  DATA:

B =              330 330 330; !B= capacity of vehicle;

    DIST=@OLE( 'M:\My Stuff\Sxoli\Diplwmatikh\liridi\mine\wrong.xls',
'dist'); !cost of route;
    P=@OLE( 'M:\My Stuff\Sxoli\Diplwmatikh\liridi\mine\wrong.xls',
'profit'); !profit for servicing nod;
    A=@OLE( 'M:\My Stuff\Sxoli\Diplwmatikh\liridi\mine\wrong.xls',
'a'); !a=demand of product;
    @OLE( 'M:\My Stuff\Sxoli\Diplwmatikh\liridi\mine\wrong.xls',
'x','y','w')=X,Y,W;

  ENDDATA


REVENUE=
@SUM(SHIP(K):
 @SUM(CITY(L):
P(L)*Y(L,K))
);
COST=
@SUM(SHIP(K):
 @SUM(CITY(L):
  @SUM(CITY(J): DIST(L,J)*X(L,J,K))
 )
);

MAX=REVENUE-COST; !objective for the PVRPP;

!assigment constraints;
@FOR(CITY(K):
 @FOR(SHIP(L):
  @SUM(CITY(J) : X(K,J,L)) = Y(K,L);
 )
);

@FOR(CITY(K):
 @FOR(SHIP(L):
```

```
  @SUM(CITY(J) : X(J,K,L)) = Y(K,L);
 )
);

!capacity contraint;
@FOR(SHIP(K) : @SUM(CITY(L) : A(L)*Y(L,K)) <= B(K););


!ensures the existence of all vehicles in the solution+a nod will not
be visited twice;
@FOR(CITY(K) | K #EQ# 1 :
 @SUM(SHIP(L) : Y(K,L)) = @SIZE(SHIP););
@FOR(CITY(K) | K #GE# 2 :
 @SUM(SHIP(L):Y(K,L))=W(K));

@FOR(LINK:@BIN(X););
@FOR(LINKZ:@BIN(Y););
@FOR(CITY(K) | K #GE# 2 :
 @BIN(W(K)));


   N=@SIZE(CITY);
!subroute elimination constraint;
@FOR(SHIP(L):
   @FOR(CITY(K):@FOR(CITY(J)|J#GT#1#AND#K#GT#1:
   U(J,L)-U(K,L) + N*X(J,K,L)<N-1;));
);
@FOR(LINKZ:U>=0;);
```

The Team Orienteering Problem

```
!code for the TOP made by Vasilis Tourloupis;

SETS:

CITY/1..18/:P,W,A;
SHIP/1..3/:B;
LINK(CITY,CITY,SHIP):X;
LINKZ(CITY,SHIP):Y,U;
LINKS(CITY,CITY):DIST;
ENDSETS

  DATA:

B =              330 330 330; !B= capacity of vehicle;

    DIST=@OLE( 'M:\My Stuff\Sxoli\Diplwmatikh\liridi\mine\wrong.xls',
'dist'); !DIST=cost of route;
    P=@OLE( 'M:\My Stuff\Sxoli\Diplwmatikh\liridi\mine\wrong.xls',
'profit'); !P=profit for servicing nod;
    A=@OLE( 'M:\My Stuff\Sxoli\Diplwmatikh\liridi\mine\wrong.xls',
'a'); !A=demand of product;
    @OLE( 'M:\My Stuff\Sxoli\Diplwmatikh\liridi\mine\wrong.xls',
'x','y','w')=X,Y,W;
```

```
  ENDDATA


REVENUE=
@SUM(SHIP(K):
 @SUM(CITY(L):
P(L)*Y(L,K))
);
COST=
@SUM(SHIP(K):
 @SUM(CITY(L):
  @SUM(CITY(J): DIST(L,J)*X(L,J,K))
 )
);

MAX=REVENUE; !objective for the TOP;
COST<=(max cost); !replace (max cost) with the actual maximum cost;

!assigment constraints;
@FOR(CITY(K):
 @FOR(SHIP(L):
  @SUM(CITY(J) : X(K,J,L)) = Y(K,L);
 )
);

@FOR(CITY(K):
 @FOR(SHIP(L):
  @SUM(CITY(J) : X(J,K,L)) = Y(K,L);
 )
);

!capacity contraint;
@FOR(SHIP(K) : @SUM(CITY(L) : A(L)*Y(L,K)) <= B(K););


!ensures the existence of all vehicles in the solution+a nod will not
be visited twice;
@FOR(CITY(K) | K #EQ# 1 :
 @SUM(SHIP(L) : Y(K,L)) = @SIZE(SHIP););
@FOR(CITY(K) | K #GE# 2 :
 @SUM(SHIP(L):Y(K,L))=W(K));

@FOR(LINK:@BIN(X););
@FOR(LINKZ:@BIN(Y););
@FOR(CITY(K) | K #GE# 2 :
 @BIN(W(K)));


   N=@SIZE(CITY);
!subroute elimination constraint;
@FOR(SHIP(L):
   @FOR(CITY(K):@FOR(CITY(J)|J#GT#1#AND#K#GT#1:
   U(J,L)-U(K,L) + N*X(J,K,L)<N-1;));
);
@FOR(LINKZ:U>=0;);
```

The Prize Collecting Vehicle Routing Problem with Profits

```
!code for the PC-VRPP made by Vasilis Tourloupis;

SETS:

CITY/1..18/:P,W,A;
SHIP/1..3/:B;
LINK(CITY,CITY,SHIP):X;
LINKZ(CITY,SHIP):Y,U;
LINKS(CITY,CITY):DIST;
ENDSETS

  DATA:

B =             330 330 330; !B= capacity of vehicle;

    DIST=@OLE( 'M:\My Stuff\Sxoli\Diplwmatikh\liridi\mine\wrong.xls',
'dist'); !DIST=cost of route;
    P=@OLE( 'M:\My Stuff\Sxoli\Diplwmatikh\liridi\mine\wrong.xls',
'profit'); !P=profit for servicing nod;
    A=@OLE( 'M:\My Stuff\Sxoli\Diplwmatikh\liridi\mine\wrong.xls',
'a'); !A=demand of product;
    @OLE( 'M:\My Stuff\Sxoli\Diplwmatikh\liridi\mine\wrong.xls',
'x','y','w')=X,Y,W;

  ENDDATA


REVENUE=
@SUM(SHIP(K):
 @SUM(CITY(L):
P(L)*Y(L,K))
);
COST=
@SUM(SHIP(K):
 @SUM(CITY(L):
  @SUM(CITY(J): DIST(L,J)*X(L,J,K))
 )
);

MIN=COST; !objective for the PC-VRPP;
REVENUE>=(min rev); !replace (min rev) with the actual minimum cost;

!assigment constraints;
@FOR(CITY(K):
 @FOR(SHIP(L):
  @SUM(CITY(J) : X(K,J,L)) = Y(K,L);
 )
);

@FOR(CITY(K):
 @FOR(SHIP(L):
  @SUM(CITY(J) : X(J,K,L)) = Y(K,L);
 )
);
```

```
!capacity contraint;
@FOR(SHIP(K) : @SUM(CITY(L) : A(L)*Y(L,K)) <= B(K););


!ensures the existence of all vehicles in the solution+a nod will not
be visited twice;
@FOR(CITY(K) | K #EQ# 1 :
 @SUM(SHIP(L) : Y(K,L)) = @SIZE(SHIP););
@FOR(CITY(K) | K #GE# 2 :
 @SUM(SHIP(L):Y(K,L))=W(K));

@FOR(LINK:@BIN(X););
@FOR(LINKZ:@BIN(Y););
@FOR(CITY(K) | K #GE# 2 :
 @BIN(W(K)));


   N=@SIZE(CITY);
!subroute elimination constraint;
@FOR(SHIP(L):
   @FOR(CITY(K):@FOR(CITY(J)|J#GT#1#AND#K#GT#1:
   U(J,L)-U(K,L) + N*X(J,K,L)<N-1;));
);
@FOR(LINKZ:U>=0;);
```

The Fleet Selective Profitable Vehicle Routing Problem with Profits

```
!code for the FS-PVRPP made by Vasilis Tourloupis;

SETS:

CITY/1..12/:P,W,A;
SHIP/1..3/:B,BUY;
LINK(CITY,CITY,SHIP):X,DIST;
LINKZ(CITY,SHIP):Y,U;
!LINKS(CITY,CITY):DIST;
ENDSETS

  DATA:

B =             315 170 675; !B= capacity of vehicle;
BUY =         138.89 153.75 156.80; !initial cost of each vehicle;
    DIST=@OLE( 'M:\Sxoli\Diplwmatikh\lyridi\mine\byshipgr12.xls',
'DIST'); !cost of route;
    P=@OLE( 'M:\Sxoli\Diplwmatikh\lyridi\mine\byshipgr12.xls',
'profit'); !profit for servicing nod;
    A=@OLE( 'M:\Sxoli\Diplwmatikh\lyridi\mine\byshipgr12.xls', 'a');
!a=demand of product;
    @OLE( 'M:\Sxoli\Diplwmatikh\lyridi\mine\byshipgr12.xls',
'x','y','w')=X,Y,W;
  ENDDATA


REVENUE=
```

```
@SUM(SHIP(K):
 @SUM(CITY(L):
P(L)*Y(L,K))
);
COST=
@SUM(SHIP(K):
 @SUM(CITY(L):
  @SUM(CITY(J): DIST(L,J,K)*X(L,J,K))
 )
);
BUYCOST=
@SUM(SHIP(K):
 @SUM(CITY(L)| L #LE# 1 : Y(L,K)*BUY(K))
);
MAX=REVENUE-COST-BUYCOST; !objective for the FS-PVRPP;

!assigment constraints;
@FOR(CITY(K):
 @FOR(SHIP(L):
  @SUM(CITY(J) : X(K,J,L)) = Y(K,L);
 )
);

@FOR(CITY(K):
 @FOR(SHIP(L):
  @SUM(CITY(J) : X(J,K,L)) = Y(K,L);
 )
);

!capacity contraint;
@FOR(SHIP(K) : @SUM(CITY(L) : A(L)*Y(L,K)) <= B(K););



!ensures a nod will not be visited twice;
@FOR(CITY(K) | K #GE# 2 :
 @SUM(SHIP(L):Y(K,L))=W(K));

@FOR(LINK:@BIN(X););
@FOR(LINKZ:@BIN(Y););
@FOR(CITY(K) | K #GE# 2 :
 @BIN(W(K)));


   N=@SIZE(CITY);
!subroute elimination constraint;
@FOR(SHIP(L):
   @FOR(CITY(K):@FOR(CITY(J)|J#GT#1#AND#K#GT#1:
   U(J,L)-U(K,L) + N*X(J,K,L)<N-1;));
);
@FOR(LINKZ:U>=0;);
```

```
!code for the FS-PC-VRPP made by Vasilis Tourloupis;

SETS:

CITY/1..12/:P,W,A;
SHIP/1..3/:B,BUY;
LINK(CITY,CITY,SHIP):X,DIST;
LINKZ(CITY,SHIP):Y,U;
!LINKS(CITY,CITY):DIST;
ENDSETS

  DATA:

B =            315 170 675; !B= capacity of vehicle;
BUY =         138.89 153.75 156.80; !initial cost of each vehicle;
    DIST=@OLE( 'M:\Sxoli\Diplwmatikh\lyridi\mine\byshipgr12.xls',
'DIST'); !cost of route;
    P=@OLE( 'M:\Sxoli\Diplwmatikh\lyridi\mine\byshipgr12.xls',
'profit'); !profit for servicing nod;
    A=@OLE( 'M:\Sxoli\Diplwmatikh\lyridi\mine\byshipgr12.xls', 'a');
!a=demand of product;
    @OLE( 'M:\Sxoli\Diplwmatikh\lyridi\mine\byshipgr12.xls',
'x','y','w')=X,Y,W;
  ENDDATA


REVENUE=
@SUM(SHIP(K):
 @SUM(CITY(L):
P(L)*Y(L,K))
);
COST=
@SUM(SHIP(K):
 @SUM(CITY(L):
  @SUM(CITY(J): DIST(L,J,K)*X(L,J,K))
 )
);
BUYCOST=          !BUYCOST=initial cost for all vehicles;
@SUM(SHIP(K):
 @SUM(CITY(L)| L #LE# 1 : Y(L,K)*BUY(K))
);

MAX=REVENUE; !objective for the TOP;
COST+BUYCOST<=(max cost); !replace (max cost) with the actual maximum
cost;

!assigment constraints;
@FOR(CITY(K):
 @FOR(SHIP(L):
  @SUM(CITY(J) : X(K,J,L)) = Y(K,L);
 )
);

@FOR(CITY(K):
```

```
 @FOR(SHIP(L):
  @SUM(CITY(J) : X(J,K,L)) = Y(K,L);
 )
);

!capacity contraint;
@FOR(SHIP(K) : @SUM(CITY(L) : A(L)*Y(L,K)) <= B(K););



!ensures a nod will not be visited twice;
@FOR(CITY(K) | K #GE# 2 :
 @SUM(SHIP(L):Y(K,L))=W(K));

@FOR(LINK:@BIN(X););
@FOR(LINKZ:@BIN(Y););
@FOR(CITY(K) | K #GE# 2 :
 @BIN(W(K)));


   N=@SIZE(CITY);
!subroute elimination constraint;
@FOR(SHIP(L):
   @FOR(CITY(K):@FOR(CITY(J)|J#GT#1#AND#K#GT#1:
   U(J,L)-U(K,L) + N*X(J,K,L)<N-1;));
);
@FOR(LINKZ:U>=0;);
```

The Fleet Selective Prize Collecting Vehicle Routing Problem with Profits

```
!code for the FS-PC-VRPP made by Vasilis Tourloupis;

SETS:

CITY/1..12/:P,W,A;
SHIP/1..3/:B,BUY;
LINK(CITY,CITY,SHIP):X,DIST;
LINKZ(CITY,SHIP):Y,U;
!LINKS(CITY,CITY):DIST;
ENDSETS

  DATA:

B =             315 170 675; !B= capacity of vehicle;
BUY =         138.89 153.75 156.80; !initial cost of each vehicle;
    DIST=@OLE( 'M:\Sxoli\Diplwmatikh\lyridi\mine\byshipgr12.xls',
'DIST'); !cost of route;
    P=@OLE( 'M:\Sxoli\Diplwmatikh\lyridi\mine\byshipgr12.xls',
'profit'); !profit for servicing nod;
    A=@OLE( 'M:\Sxoli\Diplwmatikh\lyridi\mine\byshipgr12.xls', 'a');
!a=demand of product;
    @OLE( 'M:\Sxoli\Diplwmatikh\lyridi\mine\byshipgr12.xls',
'x','y','w')=X,Y,W;
  ENDDATA
```

```
REVENUE=
@SUM(SHIP(K):
 @SUM(CITY(L):
P(L)*Y(L,K))
);
COST=
@SUM(SHIP(K):
 @SUM(CITY(L):
  @SUM(CITY(J): DIST(L,J,K)*X(L,J,K))
 )
);
BUYCOST=
@SUM(SHIP(K):
 @SUM(CITY(L)| L #LE# 1 : Y(L,K)*BUY(K))
);
MIN=COST+BUYCOST; !objective for the PC-VRPP;
REVENUE>=(min rev); !replace (min rev) with the actual minimum cost;

!assigment constraints;
@FOR(CITY(K):
 @FOR(SHIP(L):
  @SUM(CITY(J) : X(K,J,L)) = Y(K,L);
 )
);

@FOR(CITY(K):
 @FOR(SHIP(L):
  @SUM(CITY(J) : X(J,K,L)) = Y(K,L);
 )
);

!capacity contraint;
@FOR(SHIP(K) : @SUM(CITY(L) : A(L)*Y(L,K)) <= B(K););



!ensures a nod will not be visited twice;
@FOR(CITY(K) | K #GE# 2 :
 @SUM(SHIP(L):Y(K,L))=W(K));

@FOR(LINK:@BIN(X););
@FOR(LINKZ:@BIN(Y););
@FOR(CITY(K) | K #GE# 2 :
 @BIN(W(K)));


   N=@SIZE(CITY);
!subroute elimination constraint;
@FOR(SHIP(L):
   @FOR(CITY(K):@FOR(CITY(J)|J#GT#1#AND#K#GT#1:
   U(J,L)-U(K,L) + N*X(J,K,L)<N-1;));
);
@FOR(LINKZ:U>=0;);
```

Τυπικό Αποτέλεσμα LINGO

Το LINGO μας δίνει κάποιες πληροφορίες σχετικά με την διαδικασία και την πορεία της λύσης

Η στήλη reduced cost δείχνει το «κόστος» εισαγωγής της κάθε μεταβλητής στη λύση. Για παράδειγμα αν μία μεταβλητή έχει reduced cost 10, τότε η χρησιμοποίηση της στην λύση, αυτομάτως, σημαίνει μείωση του υποκειμενικού στόχου κατά 10 μονάδες

Η στήλη Slack or Surplus δείχνει το κατά πόσο κοντά είναι μία μεταβλητή να ικανοποιήσει μία ισότητα. Για παράδειγμα στην σειρά 3 έχουμε Slack or Surplus 242,42 που σημαίνει ότι το cost .με optimal value 242,42, σε αυτή τη σειρά απέχει 242,42 μονάδες από την ικανοποίηση της ισότητας της.

Η στήλη Dual Price μπορεί να ερμηνευθεί ως η ποσότητα κατά την οποία θα αυξανόταν ο υποκειμενικός στόχος εάν η σταθερή ποσότητα (δεξιά πλευρά της ισότητας) αυξανόταν κατά μία μονάδα.

Τα ονόματα των μεταβλητών είναι αυτά που έχουν δοθεί στο πρόγραμμα. Στην περίπτωσή μας είναι:

REVENUE	Τα έσοδα από την εξυπηρέτηση κόμβων
COST	Το συνολικό κόστος των διαδρομών που απαρτίζουν την λύση
N	Ο συνολικός αριθμός των κόμβων
P (i)	Τα έσοδα που θα αποφέρει η εξυπηρέτηση του κόμβου i
W(i)	Το σύνολο των οχημάτων που θα επισκεφτεί των κόμβο i
A(i)	Η ζήτηση σε προϊόν του κόμβου i
B(i)	Η μεταφορική ικανότητα του οχήματος i
X(I,j,k)	μεταβλητή απόφασης
Y(i)	1 εάν ο κόμβος I εξυπηρετείται, αλλιώς 0
U(I,k)	μεταβλητή που χρησιμοποιείται για την εξάλειψη υποδιαδρομών
DIST(I,j)	Το κόστος της διαδρομής από τον κόμβο I στον j

Ακολουθεί ένα τυπικό αποτέλεσμα του LINGO

```
Global optimal solution found.
Objective value:                 242.4200
Extended solver steps:               7810
Total solver iterations:           425193


Export Summary Report
---------------------
Transfer Method:      OLE BASED
Workbook:             M:\Sxoli\Diplwmatikh\lyridi\mine\data1.xls
Ranges Specified:       3
   x
   y
   w
Ranges Found:           1
Range Size Mismatches:      0
Values Transferred:     675
```

Variable	Value	Reduced Cost	Variable	Value	Reduced Cost	Variable	Value	Reduced Cost
REVENUE	708.8	0	X(7, 9, 3)	0	27.048	X(15, 5, 2)	0	40.204
COST	242.42	0	X(7, 10, 1)	0	4.968	X(15, 5, 3)	0	40.204
N	15	0	X(7, 10, 2)	0	4.968	X(15, 6, 1)	0	52.44
P(1)	0	0	X(7, 10, 3)	1	4.968	X(15, 6, 2)	0	52.44
P(2)	355.2	0	X(7, 11, 1)	0	35.788	X(15, 6, 3)	0	52.44
P(3)	82.4	0	X(7, 11, 2)	0	35.788	X(15, 7, 1)	0	62.284
P(4)	61.6	0	X(7, 11, 3)	0	35.788	X(15, 7, 2)	0	62.284
P(5)	54.4	0	X(7, 12, 1)	0	25.3	X(15, 7, 3)	0	62.284
P(6)	34.4	0	X(7, 12, 2)	0	25.3	X(15, 8, 1)	0	5.152
P(7)	28.8	0	X(7, 12, 3)	0	25.3	X(15, 8, 2)	0	5.152
P(8)	24.8	0	X(7, 13, 1)	0	14.444	X(15, 8, 3)	0	5.152
P(9)	24.8	0	X(7, 13, 2)	0	14.444	X(15, 9, 1)	0	85.836
P(10)	24	0	X(7, 13, 3)	0	14.444	X(15, 9, 2)	0	85.836
P(11)	23.2	0	X(7, 14, 1)	0	73.784	X(15, 9, 3)	0	85.836
P(12)	23.2	0	X(7, 14, 2)	0	73.784	X(15, 10, 1)	0	14.26
P(13)	21.6	0	X(7, 14, 3)	0	73.784	X(15, 10, 2)	0	14.26
P(14)	20.8	0	X(7, 15, 1)	0	62.284	X(15, 10, 3)	0	14.26
P(15)	20	0	X(7, 15, 2)	0	62.284	X(15, 11, 1)	0	26.772
W(1)	0	0	X(7, 15, 3)	0	62.284	X(15, 11, 2)	0	26.772
W(2)	1	0	X(8, 1, 1)	0	62.56	X(15, 11, 3)	0	26.772
W(3)	1	0	X(8, 1, 2)	0	62.56	X(15, 12, 1)	0	40.572
W(4)	1	0	X(8, 1, 3)	0	62.56	X(15, 12, 2)	0	40.572
W(5)	1	0	X(8, 2, 1)	0	15.18	X(15, 12, 3)	0	40.572
W(6)	1	0	X(8, 2, 2)	0	15.18	X(15, 13, 1)	0	47.932
W(7)	1	0	X(8, 2, 3)	0	15.18	X(15, 13, 2)	0	47.932
W(8)	0	0	X(8, 3, 1)	0	57.408	X(15, 13, 3)	0	47.932
W(9)	0	0	X(8, 3, 2)	0	57.408	X(15, 14, 1)	0	11.224
W(10)	1	0	X(8, 3, 3)	0	57.408	X(15, 14, 2)	0	11.224
W(11)	1	0	X(8, 4, 1)	0	32.292	X(15, 14, 3)	0	11.224
W(12)	1	0	X(8, 4, 2)	0	32.292	X(15, 15, 1)	0	460
W(13)	1	0	X(8, 4, 3)	0	32.292	X(15, 15, 2)	0	460
W(14)	0	0	X(8, 5, 1)	0	35.236	X(15, 15, 3)	0	460
W(15)	0	0	X(8, 5, 2)	0	35.236	Y(1, 1)	1	0
A(1)	0	0	X(8, 5, 3)	0	35.236	Y(1, 2)	1	0
A(2)	444	0	X(8, 6, 1)	0	49.22	Y(1, 3)	1	0
A(3)	103	0	X(8, 6, 2)	0	49.22	Y(2, 1)	0	0
A(4)	77	0	X(8, 6, 3)	0	49.22	Y(2, 2)	1	0
A(5)	68	0	X(8, 7, 1)	0	57.132	Y(2, 3)	0	0
A(6)	43	0	X(8, 7, 2)	0	57.132	Y(3, 1)	0	0
A(7)	36	0	X(8, 7, 3)	0	57.132	Y(3, 2)	0	0
A(8)	31	0	X(8, 8, 1)	0	460	Y(3, 3)	1	0
A(9)	31	0	X(8, 8, 2)	0	460	Y(4, 1)	0	0
A(10)	30	0	X(8, 8, 3)	0	460	Y(4, 2)	0	0
A(11)	29	0	X(8, 9, 1)	0	80.776	Y(4, 3)	1	0
A(12)	29	0	X(8, 9, 2)	0	80.776	Y(5, 1)	0	0
A(13)	27	0	X(8, 9, 3)	0	80.776	Y(5, 2)	0	0
A(14)	26	0	X(8, 10, 1)	0	9.752	Y(5, 3)	1	0
A(15)	25	0	X(8, 10, 2)	0	9.752	Y(6, 1)	0	0
B(1)	450	0	X(8, 10, 3)	0	9.752	Y(6, 2)	0	0
B(2)	450	0	X(8, 11, 1)	0	24.472	Y(6, 3)	1	0
B(3)	450	0	X(8, 11, 2)	0	24.472	Y(7, 1)	0	0
X(1, 1, 1)	0	460	X(8, 11, 3)	0	24.472	Y(7, 2)	0	0
X(1, 1, 2)	0	460	X(8, 12, 1)	0	35.42	Y(7, 3)	1	0
X(1, 1, 3)	0	460	X(8, 12, 2)	0	35.42	Y(8, 1)	0	0
X(1, 2, 1)	0	47.38	X(8, 12, 3)	0	35.42	Y(8, 2)	0	0
X(1, 2, 2)	1	47.38	X(8, 13, 1)	0	45.724	Y(8, 3)	0	0
X(1, 2, 3)	0	47.38	X(8, 13, 2)	0	45.724	Y(9, 1)	0	0
X(1, 3, 1)	0	19.872	X(8, 13, 3)	0	45.724	Y(9, 2)	0	0
X(1, 3, 2)	0	19.872	X(8, 14, 1)	0	16.284	Y(9, 3)	0	0
X(1, 3, 3)	0	19.872	X(8, 14, 2)	0	16.284	Y(10, 1)	0	0
X(1, 4, 1)	0	33.028	X(8, 14, 3)	0	16.284	Y(10, 2)	0	0
X(1, 4, 2)	0	33.028	X(8, 15, 1)	0	5.152	Y(10, 3)	1	0
X(1, 4, 3)	0	33.028	X(8, 15, 2)	0	5.152	Y(11, 1)	0	0
X(1, 5, 1)	0	29.808	X(8, 15, 3)	0	5.152	Y(11, 2)	0	0
X(1, 5, 2)	0	29.808	X(9, 1, 1)	0	26.128	Y(11, 3)	1	0
X(1, 5, 3)	0	29.808	X(9, 1, 2)	0	26.128	Y(12, 1)	0	0
X(1, 6, 1)	0	40.296	X(9, 1, 3)	0	26.128	Y(12, 2)	0	0
X(1, 6, 2)	0	40.296	X(9, 2, 1)	0	65.78	Y(12, 3)	1	0
X(1, 6, 3)	0	40.296	X(9, 2, 2)	0	65.78	Y(13, 1)	1	0
X(1, 7, 1)	0	8.096	X(9, 2, 3)	0	65.78	Y(13, 2)	0	0
X(1, 7, 2)	0	8.096	X(9, 3, 1)	0	19.78	Y(13, 3)	0	0
X(1, 7, 3)	1	8.096	X(9, 3, 2)	0	19.78	Y(14, 1)	0	0
X(1, 8, 1)	0	62.56	X(9, 3, 3)	0	19.78	Y(14, 2)	0	0
X(1, 8, 2)	0	62.56	X(9, 4, 1)	0	51.796	Y(14, 3)	0	0
X(1, 8, 3)	0	62.56	X(9, 4, 2)	0	51.796	Y(15, 1)	0	0
X(1, 9, 1)	0	26.128	X(9, 4, 3)	0	51.796	Y(15, 2)	0	0
X(1, 9, 2)	0	26.128	X(9, 5, 1)	0	48.392	Y(15, 3)	0	0
X(1, 9, 3)	0	26.128	X(9, 5, 2)	0	48.392	U(1, 1)	0	0
X(1, 10, 1)	0	56.304	X(9, 5, 3)	0	48.392	U(1, 2)	0	0
X(1, 10, 2)	0	56.304	X(9, 6, 1)	0	42.044	U(1, 3)	0	0
X(1, 10, 3)	0	56.304	X(9, 6, 2)	0	42.044	U(2, 1)	0	0
X(1, 11, 1)	0	41.032	X(9, 6, 3)	0	42.044	U(2, 2)	0	0
X(1, 11, 2)	0	41.032	X(9, 7, 1)	0	27.048	U(2, 3)	0	0
X(1, 11, 3)	0	41.032	X(9, 7, 2)	0	27.048	U(3, 1)	0	0
X(1, 12, 1)	0	30.452	X(9, 7, 3)	0	27.048	U(3, 2)	0	0
X(1, 12, 2)	0	30.452	X(9, 8, 1)	0	80.776	U(3, 3)	14	0
X(1, 12, 3)	0	30.452	X(9, 8, 2)	0	80.776	U(4, 1)	0	0
X(1, 13, 1)	1	19.688	X(9, 8, 3)	0	80.776	U(4, 2)	14	0

X(1, 13, 2)	0	19.688	X(9, 9, 1)	0	460	U(4, 3)	10	0
X(1, 13, 3)	0	19.688	X(9, 9, 2)	0	460	U(5, 1)	14	0
X(1, 14, 1)	0	78.568	X(9, 9, 3)	0	460	U(5, 2)	0	0
X(1, 14, 2)	0	78.568	X(9, 10, 1)	0	73.14	U(5, 3)	11	0
X(1, 14, 3)	0	78.568	X(9, 10, 2)	0	73.14	U(6, 1)	14	0
X(1, 15, 1)	0	67.712	X(9, 10, 3)	0	73.14	U(6, 2)	0	0
X(1, 15, 2)	0	67.712	X(9, 11, 1)	0	59.432	U(6, 3)	13	0
X(1, 15, 3)	0	67.712	X(9, 11, 2)	0	59.432	U(7, 1)	14	0
X(2, 1, 1)	0	47.38	X(9, 11, 3)	0	59.432	U(7, 2)	14	0
X(2, 1, 2)	1	47.38	X(9, 12, 1)	0	48.852	U(7, 3)	0	0
X(2, 1, 3)	0	47.38	X(9, 12, 2)	0	48.852	U(8, 1)	14	0
X(2, 2, 1)	0	460	X(9, 12, 3)	0	48.852	U(8, 2)	14	0
X(2, 2, 2)	0	460	X(9, 13, 1)	0	38.18	U(8, 3)	0	0
X(2, 2, 3)	0	460	X(9, 13, 2)	0	38.18	U(9, 1)	0	0
X(2, 3, 1)	0	42.228	X(9, 13, 3)	0	38.18	U(9, 2)	0	0
X(2, 3, 2)	0	42.228	X(9, 14, 1)	0	97.06	U(9, 3)	0	0
X(2, 3, 3)	0	42.228	X(9, 14, 2)	0	97.06	U(10, 1)	14	0
X(2, 4, 1)	0	17.112	X(9, 14, 3)	0	97.06	U(10, 2)	14	0
X(2, 4, 2)	0	17.112	X(9, 15, 1)	0	85.836	U(10, 3)	1	0
X(2, 4, 3)	0	17.112	X(9, 15, 2)	0	85.836	U(11, 1)	14	0
X(2, 5, 1)	0	19.872	X(9, 15, 3)	0	85.836	U(11, 2)	14	0
X(2, 5, 2)	0	19.872	X(10, 1, 1)	0	56.304	U(11, 3)	2	0
X(2, 5, 3)	0	19.872	X(10, 1, 2)	0	56.304	U(12, 1)	14	0
X(2, 6, 1)	0	34.04	X(10, 1, 3)	0	56.304	U(12, 2)	0	0
X(2, 6, 2)	0	34.04	X(10, 2, 1)	0	7.82	U(12, 3)	12	0
X(2, 6, 3)	0	34.04	X(10, 2, 2)	0	7.82	U(13, 1)	0	0
X(2, 7, 1)	0	41.952	X(10, 2, 3)	0	7.82	U(13, 2)	0	0
X(2, 7, 2)	0	41.952	X(10, 3, 1)	0	53.452	U(13, 3)	14	0
X(2, 7, 3)	0	41.952	X(10, 3, 2)	0	53.452	U(14, 1)	0	0
X(2, 8, 1)	0	15.18	X(10, 3, 3)	0	53.452	U(14, 2)	0	0
X(2, 8, 2)	0	15.18	X(10, 4, 1)	0	22.356	U(14, 3)	0	0
X(2, 8, 3)	0	15.18	X(10, 4, 2)	0	22.356	U(15, 1)	14	0
X(2, 9, 1)	0	65.78	X(10, 4, 3)	0	22.356	U(15, 2)	14	0
X(2, 9, 2)	0	65.78	X(10, 5, 1)	0	27.784	U(15, 3)	0	0
X(2, 9, 3)	0	65.78	X(10, 5, 2)	0	27.784	DIST(1, 1)	460	0
X(2, 10, 1)	0	7.82	X(10, 5, 3)	0	27.784	DIST(1, 2)	47.38	0
X(2, 10, 2)	0	7.82	X(10, 6, 1)	0	39.928	DIST(1, 3)	19.872	0
X(2, 10, 3)	0	7.82	X(10, 6, 2)	0	39.928	DIST(1, 4)	33.028	0
X(2, 11, 1)	0	9.292	X(10, 6, 3)	0	39.928	DIST(1, 5)	29.808	0
X(2, 11, 2)	0	9.292	X(10, 7, 1)	0	4.968	DIST(1, 6)	40.296	0
X(2, 11, 3)	0	9.292	X(10, 7, 2)	0	4.968	DIST(1, 7)	8.096	0
X(2, 12, 1)	0	20.24	X(10, 7, 3)	0	4.968	DIST(1, 8)	62.56	0
X(2, 12, 2)	0	20.24	X(10, 8, 1)	0	9.752	DIST(1, 9)	26.128	0
X(2, 12, 3)	0	20.24	X(10, 8, 2)	0	9.752	DIST(1, 10)	56.304	0
X(2, 13, 1)	0	30.544	X(10, 8, 3)	0	9.752	DIST(1, 11)	41.032	0
X(2, 13, 2)	0	30.544	X(10, 9, 1)	0	73.14	DIST(1, 12)	30.452	0
X(2, 13, 3)	0	30.544	X(10, 9, 2)	0	73.14	DIST(1, 13)	19.688	0
X(2, 14, 1)	0	31.832	X(10, 9, 3)	0	73.14	DIST(1, 14)	78.568	0
X(2, 14, 2)	0	31.832	X(10, 10, 1)	0	460	DIST(1, 15)	67.712	0
X(2, 14, 3)	0	31.832	X(10, 10, 2)	0	460	DIST(2, 1)	47.38	0
X(2, 15, 1)	0	20.332	X(10, 10, 3)	0	460	DIST(2, 2)	460	0
X(2, 15, 2)	0	20.332	X(10, 11, 1)	0	14.26	DIST(2, 3)	42.228	0
X(2, 15, 3)	0	20.332	X(10, 11, 2)	0	14.26	DIST(2, 4)	17.112	0
X(3, 1, 1)	0	19.872	X(10, 11, 3)	1	14.26	DIST(2, 5)	19.872	0
X(3, 1, 2)	0	19.872	X(10, 12, 1)	0	28.06	DIST(2, 6)	34.04	0
X(3, 1, 3)	1	19.872	X(10, 12, 2)	0	28.06	DIST(2, 7)	41.952	0
X(3, 2, 1)	0	42.228	X(10, 12, 3)	0	28.06	DIST(2, 8)	15.18	0
X(3, 2, 2)	0	42.228	X(10, 13, 1)	0	35.42	DIST(2, 9)	65.78	0
X(3, 2, 3)	0	42.228	X(10, 13, 2)	0	35.42	DIST(2, 10)	7.82	0
X(3, 3, 1)	0	460	X(10, 13, 3)	0	35.42	DIST(2, 11)	9.292	0
X(3, 3, 2)	0	460	X(10, 14, 1)	0	24.932	DIST(2, 12)	20.24	0
X(3, 3, 3)	0	460	X(10, 14, 2)	0	24.932	DIST(2, 13)	30.544	0
X(3, 4, 1)	0	31.096	X(10, 14, 3)	0	24.932	DIST(2, 14)	31.832	0
X(3, 4, 2)	0	31.096	X(10, 15, 1)	0	14.26	DIST(2, 15)	20.332	0
X(3, 4, 3)	0	31.096	X(10, 15, 2)	0	14.26	DIST(3, 1)	19.872	0
X(3, 5, 1)	0	28.796	X(10, 15, 3)	0	14.26	DIST(3, 2)	42.228	0
X(3, 5, 2)	0	28.796	X(11, 1, 1)	0	41.032	DIST(3, 3)	460	0
X(3, 5, 3)	0	28.796	X(11, 1, 2)	0	41.032	DIST(3, 4)	31.096	0
X(3, 6, 1)	0	22.54	X(11, 1, 3)	0	41.032	DIST(3, 5)	28.796	0
X(3, 6, 2)	0	22.54	X(11, 2, 1)	0	9.292	DIST(3, 6)	22.54	0
X(3, 6, 3)	0	22.54	X(11, 2, 2)	0	9.292	DIST(3, 7)	26.036	0
X(3, 7, 1)	0	26.036	X(11, 2, 3)	0	9.292	DIST(3, 8)	57.408	0
X(3, 7, 2)	0	26.036	X(11, 3, 1)	0	39.192	DIST(3, 9)	19.78	0
X(3, 7, 3)	0	26.036	X(11, 3, 2)	0	39.192	DIST(3, 10)	53.452	0
X(3, 8, 1)	0	57.408	X(11, 3, 3)	0	39.192	DIST(3, 11)	39.192	0
X(3, 8, 2)	0	57.408	X(11, 4, 1)	0	8.096	DIST(3, 12)	28.888	0
X(3, 8, 3)	0	57.408	X(11, 4, 2)	0	8.096	DIST(3, 13)	18.032	0
X(3, 9, 1)	0	19.78	X(11, 4, 3)	1	8.096	DIST(3, 14)	76.176	0
X(3, 9, 2)	0	19.78	X(11, 5, 1)	0	13.708	DIST(3, 15)	65.964	0
X(3, 9, 3)	0	19.78	X(11, 5, 2)	0	13.708	DIST(4, 1)	33.028	0
X(3, 10, 1)	0	53.452	X(11, 5, 3)	0	13.708	DIST(4, 2)	17.112	0
X(3, 10, 2)	0	53.452	X(11, 6, 1)	0	27.324	DIST(4, 3)	31.096	0
X(3, 10, 3)	0	53.452	X(11, 6, 2)	0	27.324	DIST(4, 4)	460	0
X(3, 11, 1)	0	39.192	X(11, 6, 3)	0	27.324	DIST(4, 5)	5.52	0
X(3, 11, 2)	0	39.192	X(11, 7, 1)	0	35.788	DIST(4, 6)	19.504	0
X(3, 11, 3)	0	39.192	X(11, 7, 2)	0	35.788	DIST(4, 7)	27.232	0
X(3, 12, 1)	0	28.888	X(11, 7, 3)	0	35.788	DIST(4, 8)	32.292	0
X(3, 12, 2)	0	28.888	X(11, 8, 1)	0	24.472	DIST(4, 9)	51.796	0
X(3, 12, 3)	0	28.888	X(11, 8, 2)	0	24.472	DIST(4, 10)	22.356	0
X(3, 13, 1)	0	18.032	X(11, 8, 3)	0	24.472	DIST(4, 11)	8.096	0

X(3, 13, 2)	0	18.032	X(11, 9, 1)	0	59.432	DIST(4, 12)	5.704	0
X(3, 13, 3)	0	18.032	X(11, 9, 2)	0	59.432	DIST(4, 13)	13.616	0
X(3, 14, 1)	0	76.176	X(11, 9, 3)	0	59.432	DIST(4, 14)	45.356	0
X(3, 14, 2)	0	76.176	X(11, 10, 1)	0	14.26	DIST(4, 15)	34.868	0
X(3, 14, 3)	0	76.176	X(11, 10, 2)	0	14.26	DIST(5, 1)	29.808	0
X(3, 15, 1)	0	65.964	X(11, 10, 3)	0	14.26	DIST(5, 2)	19.872	0
X(3, 15, 2)	0	65.964	X(11, 11, 1)	0	460	DIST(5, 3)	28.796	0
X(3, 15, 3)	0	65.964	X(11, 11, 2)	0	460	DIST(5, 4)	5.52	0
X(4, 1, 1)	0	33.028	X(11, 11, 3)	0	460	DIST(5, 5)	460	0
X(4, 1, 2)	0	33.028	X(11, 12, 1)	0	13.984	DIST(5, 6)	25.208	0
X(4, 1, 3)	0	33.028	X(11, 12, 2)	0	13.984	DIST(5, 7)	24.472	0
X(4, 2, 1)	0	17.112	X(11, 12, 3)	0	13.984	DIST(5, 8)	35.236	0
X(4, 2, 2)	0	17.112	X(11, 13, 1)	0	21.16	DIST(5, 9)	48.392	0
X(4, 2, 3)	0	17.112	X(11, 13, 2)	0	21.16	DIST(5, 10)	27.784	0
X(4, 3, 1)	0	31.096	X(11, 13, 3)	0	21.16	DIST(5, 11)	13.708	0
X(4, 3, 2)	0	31.096	X(11, 14, 1)	0	37.812	DIST(5, 12)	11.132	0
X(4, 3, 3)	0	31.096	X(11, 14, 2)	0	37.812	DIST(5, 13)	10.58	0
X(4, 4, 1)	0	460	X(11, 14, 3)	0	37.812	DIST(5, 14)	51.152	0
X(4, 4, 2)	0	460	X(11, 15, 1)	0	26.772	DIST(5, 15)	40.204	0
X(4, 4, 3)	0	460	X(11, 15, 2)	0	26.772	DIST(6, 1)	40.296	0
X(4, 5, 1)	0	5.52	X(11, 15, 3)	0	26.772	DIST(6, 2)	34.04	0
X(4, 5, 2)	0	5.52	X(12, 1, 1)	0	30.452	DIST(6, 3)	22.54	0
X(4, 5, 3)	1	5.52	X(12, 1, 2)	0	30.452	DIST(6, 4)	19.504	0
X(4, 6, 1)	0	19.504	X(12, 1, 3)	0	30.452	DIST(6, 5)	25.208	0
X(4, 6, 2)	0	19.504	X(12, 2, 1)	0	20.24	DIST(6, 6)	460	0
X(4, 6, 3)	0	19.504	X(12, 2, 2)	0	20.24	DIST(6, 7)	38.916	0
X(4, 7, 1)	0	27.232	X(12, 2, 3)	0	20.24	DIST(6, 8)	49.22	0
X(4, 7, 2)	0	27.232	X(12, 3, 1)	0	28.888	DIST(6, 9)	42.044	0
X(4, 7, 3)	0	27.232	X(12, 3, 2)	0	28.888	DIST(6, 10)	39.928	0
X(4, 8, 1)	0	32.292	X(12, 3, 3)	0	28.888	DIST(6, 11)	27.324	0
X(4, 8, 2)	0	32.292	X(12, 4, 1)	0	5.704	DIST(6, 12)	13.8	0
X(4, 8, 3)	0	32.292	X(12, 4, 2)	0	5.704	DIST(6, 13)	24.564	0
X(4, 9, 1)	0	51.796	X(12, 4, 3)	0	5.704	DIST(6, 14)	64.584	0
X(4, 9, 2)	0	51.796	X(12, 5, 1)	0	11.132	DIST(6, 15)	52.44	0
X(4, 9, 3)	0	51.796	X(12, 5, 2)	0	11.132	DIST(7, 1)	8.096	0
X(4, 10, 1)	0	22.356	X(12, 5, 3)	0	11.132	DIST(7, 2)	41.952	0
X(4, 10, 2)	0	22.356	X(12, 6, 1)	0	13.8	DIST(7, 3)	26.036	0
X(4, 10, 3)	0	22.356	X(12, 6, 2)	0	13.8	DIST(7, 4)	27.232	0
X(4, 11, 1)	0	8.096	X(12, 6, 3)	1	13.8	DIST(7, 5)	24.472	0
X(4, 11, 2)	0	8.096	X(12, 7, 1)	0	25.3	DIST(7, 6)	38.916	0
X(4, 11, 3)	0	8.096	X(12, 7, 2)	0	25.3	DIST(7, 7)	460	0
X(4, 12, 1)	0	5.704	X(12, 7, 3)	0	25.3	DIST(7, 8)	57.132	0
X(4, 12, 2)	0	5.704	X(12, 8, 1)	0	35.42	DIST(7, 9)	27.048	0
X(4, 12, 3)	0	5.704	X(12, 8, 2)	0	35.42	DIST(7, 10)	4.968	0
X(4, 13, 1)	0	13.616	X(12, 8, 3)	0	35.42	DIST(7, 11)	35.788	0
X(4, 13, 2)	0	13.616	X(12, 9, 1)	0	48.852	DIST(7, 12)	25.3	0
X(4, 13, 3)	0	13.616	X(12, 9, 2)	0	48.852	DIST(7, 13)	14.444	0
X(4, 14, 1)	0	45.356	X(12, 9, 3)	0	48.852	DIST(7, 14)	73.784	0
X(4, 14, 2)	0	45.356	X(12, 10, 1)	0	28.06	DIST(7, 15)	62.284	0
X(4, 14, 3)	0	45.356	X(12, 10, 2)	0	28.06	DIST(8, 1)	62.56	0
X(4, 15, 1)	0	34.868	X(12, 10, 3)	0	28.06	DIST(8, 2)	15.18	0
X(4, 15, 2)	0	34.868	X(12, 11, 1)	0	13.984	DIST(8, 3)	57.408	0
X(4, 15, 3)	0	34.868	X(12, 11, 2)	0	13.984	DIST(8, 4)	32.292	0
X(5, 1, 1)	0	29.808	X(12, 11, 3)	0	13.984	DIST(8, 5)	35.236	0
X(5, 1, 2)	0	29.808	X(12, 12, 1)	0	460	DIST(8, 6)	49.22	0
X(5, 1, 3)	0	29.808	X(12, 12, 2)	0	460	DIST(8, 7)	57.132	0
X(5, 2, 1)	0	19.872	X(12, 12, 3)	0	460	DIST(8, 8)	460	0
X(5, 2, 2)	0	19.872	X(12, 13, 1)	0	10.856	DIST(8, 9)	80.776	0
X(5, 2, 3)	0	19.872	X(12, 13, 2)	0	10.856	DIST(8, 10)	9.752	0
X(5, 3, 1)	0	28.796	X(12, 13, 3)	0	10.856	DIST(8, 11)	24.472	0
X(5, 3, 2)	0	28.796	X(12, 14, 1)	0	50.968	DIST(8, 12)	35.42	0
X(5, 3, 3)	0	28.796	X(12, 14, 2)	0	50.968	DIST(8, 13)	45.724	0
X(5, 4, 1)	0	5.52	X(12, 14, 3)	0	50.968	DIST(8, 14)	16.284	0
X(5, 4, 2)	0	5.52	X(12, 15, 1)	0	40.572	DIST(8, 15)	5.152	0
X(5, 4, 3)	0	5.52	X(12, 15, 2)	0	40.572	DIST(9, 1)	26.128	0
X(5, 5, 1)	0	460	X(12, 15, 3)	0	40.572	DIST(9, 2)	65.78	0
X(5, 5, 2)	0	460	X(13, 1, 1)	1	19.688	DIST(9, 3)	19.78	0
X(5, 5, 3)	0	460	X(13, 1, 2)	0	19.688	DIST(9, 4)	51.796	0
X(5, 6, 1)	0	25.208	X(13, 1, 3)	0	19.688	DIST(9, 5)	48.392	0
X(5, 6, 2)	0	25.208	X(13, 2, 1)	0	30.544	DIST(9, 6)	42.044	0
X(5, 6, 3)	0	25.208	X(13, 2, 2)	0	30.544	DIST(9, 7)	27.048	0
X(5, 7, 1)	0	24.472	X(13, 2, 3)	0	30.544	DIST(9, 8)	80.776	0
X(5, 7, 2)	0	24.472	X(13, 3, 1)	0	18.032	DIST(9, 9)	460	0
X(5, 7, 3)	0	24.472	X(13, 3, 2)	0	18.032	DIST(9, 10)	73.14	0
X(5, 8, 1)	0	35.236	X(13, 3, 3)	0	18.032	DIST(9, 11)	59.432	0
X(5, 8, 2)	0	35.236	X(13, 4, 1)	0	13.616	DIST(9, 12)	48.852	0
X(5, 8, 3)	0	35.236	X(13, 4, 2)	0	13.616	DIST(9, 13)	38.18	0
X(5, 9, 1)	0	48.392	X(13, 4, 3)	0	13.616	DIST(9, 14)	97.06	0
X(5, 9, 2)	0	48.392	X(13, 5, 1)	0	10.58	DIST(9, 15)	85.836	0
X(5, 9, 3)	0	48.392	X(13, 5, 2)	0	10.58	DIST(10, 1)	56.304	0
X(5, 10, 1)	0	27.784	X(13, 5, 3)	0	10.58	DIST(10, 2)	7.82	0
X(5, 10, 2)	0	27.784	X(13, 6, 1)	0	24.564	DIST(10, 3)	53.452	0
X(5, 10, 3)	0	27.784	X(13, 6, 2)	0	24.564	DIST(10, 4)	22.356	0
X(5, 11, 1)	0	13.708	X(13, 6, 3)	0	24.564	DIST(10, 5)	27.784	0
X(5, 11, 2)	0	13.708	X(13, 7, 1)	0	14.444	DIST(10, 6)	39.928	0
X(5, 11, 3)	0	13.708	X(13, 7, 2)	0	14.444	DIST(10, 7)	4.968	0
X(5, 12, 1)	0	11.132	X(13, 7, 3)	0	14.444	DIST(10, 8)	9.752	0
X(5, 12, 2)	0	11.132	X(13, 8, 1)	0	45.724	DIST(10, 9)	73.14	0
X(5, 12, 3)	1	11.132	X(13, 8, 2)	0	45.724	DIST(10, 10)	460	0
X(5, 13, 1)	0	10.58	X(13, 8, 3)	0	45.724	DIST(10, 11)	14.26	0

X(5, 13, 2)	0	10.58	X(13, 9, 1)	0	38.18	DIST(10, 12)	28.06	0
X(5, 13, 3)	0	10.58	X(13, 9, 2)	0	38.18	DIST(10, 13)	35.42	0
X(5, 14, 1)	0	51.152	X(13, 9, 3)	0	38.18	DIST(10, 14)	24.932	0
X(5, 14, 2)	0	51.152	X(13, 10, 1)	0	35.42	DIST(10, 15)	14.26	0
X(5, 14, 3)	0	51.152	X(13, 10, 2)	0	35.42	DIST(11, 1)	41.032	0
X(5, 15, 1)	0	40.204	X(13, 10, 3)	0	35.42	DIST(11, 2)	9.292	0
X(5, 15, 2)	0	40.204	X(13, 11, 1)	0	21.16	DIST(11, 3)	39.192	0
X(5, 15, 3)	0	40.204	X(13, 11, 2)	0	21.16	DIST(11, 4)	8.096	0
X(6, 1, 1)	0	40.296	X(13, 11, 3)	0	21.16	DIST(11, 5)	13.708	0
X(6, 1, 2)	0	40.296	X(13, 12, 1)	0	10.856	DIST(11, 6)	27.324	0
X(6, 1, 3)	0	40.296	X(13, 12, 2)	0	10.856	DIST(11, 7)	35.788	0
X(6, 2, 1)	0	34.04	X(13, 12, 3)	0	10.856	DIST(11, 8)	24.472	0
X(6, 2, 2)	0	34.04	X(13, 13, 1)	0	460	DIST(11, 9)	59.432	0
X(6, 2, 3)	0	34.04	X(13, 13, 2)	0	460	DIST(11, 10)	14.26	0
X(6, 3, 1)	0	22.54	X(13, 13, 3)	0	460	DIST(11, 11)	460	0
X(6, 3, 2)	0	22.54	X(13, 14, 1)	0	59.156	DIST(11, 12)	13.984	0
X(6, 3, 3)	1	22.54	X(13, 14, 2)	0	59.156	DIST(11, 13)	21.16	0
X(6, 4, 1)	0	19.504	X(13, 14, 3)	0	59.156	DIST(11, 14)	37.812	0
X(6, 4, 2)	0	19.504	X(13, 15, 1)	0	47.932	DIST(11, 15)	26.772	0
X(6, 4, 3)	0	19.504	X(13, 15, 2)	0	47.932	DIST(12, 1)	30.452	0
X(6, 5, 1)	0	25.208	X(13, 15, 3)	0	47.932	DIST(12, 2)	20.24	0
X(6, 5, 2)	0	25.208	X(14, 1, 1)	0	78.568	DIST(12, 3)	28.888	0
X(6, 5, 3)	0	25.208	X(14, 1, 2)	0	78.568	DIST(12, 4)	5.704	0
X(6, 6, 1)	0	460	X(14, 1, 3)	0	78.568	DIST(12, 5)	11.132	0
X(6, 6, 2)	0	460	X(14, 2, 1)	0	31.832	DIST(12, 6)	13.8	0
X(6, 6, 3)	0	460	X(14, 2, 2)	0	31.832	DIST(12, 7)	25.3	0
X(6, 7, 1)	0	38.916	X(14, 2, 3)	0	31.832	DIST(12, 8)	35.42	0
X(6, 7, 2)	0	38.916	X(14, 3, 1)	0	76.176	DIST(12, 9)	48.852	0
X(6, 7, 3)	0	38.916	X(14, 3, 2)	0	76.176	DIST(12, 10)	28.06	0
X(6, 8, 1)	0	49.22	X(14, 3, 3)	0	76.176	DIST(12, 11)	13.984	0
X(6, 8, 2)	0	49.22	X(14, 4, 1)	0	45.356	DIST(12, 12)	460	0
X(6, 8, 3)	0	49.22	X(14, 4, 2)	0	45.356	DIST(12, 13)	10.856	0
X(6, 9, 1)	0	42.044	X(14, 4, 3)	0	45.356	DIST(12, 14)	50.968	0
X(6, 9, 2)	0	42.044	X(14, 5, 1)	0	51.152	DIST(12, 15)	40.572	0
X(6, 9, 3)	0	42.044	X(14, 5, 2)	0	51.152	DIST(13, 1)	19.688	0
X(6, 10, 1)	0	39.928	X(14, 5, 3)	0	51.152	DIST(13, 2)	30.544	0
X(6, 10, 2)	0	39.928	X(14, 6, 1)	0	64.584	DIST(13, 3)	18.032	0
X(6, 10, 3)	0	39.928	X(14, 6, 2)	0	64.584	DIST(13, 4)	13.616	0
X(6, 11, 1)	0	27.324	X(14, 6, 3)	0	64.584	DIST(13, 5)	10.58	0
X(6, 11, 2)	0	27.324	X(14, 7, 1)	0	73.784	DIST(13, 6)	24.564	0
X(6, 11, 3)	0	27.324	X(14, 7, 2)	0	73.784	DIST(13, 7)	14.444	0
X(6, 12, 1)	0	13.8	X(14, 7, 3)	0	73.784	DIST(13, 8)	45.724	0
X(6, 12, 2)	0	13.8	X(14, 8, 1)	0	16.284	DIST(13, 9)	38.18	0
X(6, 12, 3)	0	13.8	X(14, 8, 2)	0	16.284	DIST(13, 10)	35.42	0
X(6, 13, 1)	0	24.564	X(14, 8, 3)	0	16.284	DIST(13, 11)	21.16	0
X(6, 13, 2)	0	24.564	X(14, 9, 1)	0	97.06	DIST(13, 12)	10.856	0
X(6, 13, 3)	0	24.564	X(14, 9, 2)	0	97.06	DIST(13, 13)	460	0
X(6, 14, 1)	0	64.584	X(14, 9, 3)	0	97.06	DIST(13, 14)	59.156	0
X(6, 14, 2)	0	64.584	X(14, 10, 1)	0	24.932	DIST(13, 15)	47.932	0
X(6, 14, 3)	0	64.584	X(14, 10, 2)	0	24.932	DIST(14, 1)	78.568	0
X(6, 15, 1)	0	52.44	X(14, 10, 3)	0	24.932	DIST(14, 2)	31.832	0
X(6, 15, 2)	0	52.44	X(14, 11, 1)	0	37.812	DIST(14, 3)	76.176	0
X(6, 15, 3)	0	52.44	X(14, 11, 2)	0	37.812	DIST(14, 4)	45.356	0
X(7, 1, 1)	0	8.096	X(14, 11, 3)	0	37.812	DIST(14, 5)	51.152	0
X(7, 1, 2)	0	8.096	X(14, 12, 1)	0	50.968	DIST(14, 6)	64.584	0
X(7, 1, 3)	0	8.096	X(14, 12, 2)	0	50.968	DIST(14, 7)	73.784	0
X(7, 2, 1)	0	41.952	X(14, 12, 3)	0	50.968	DIST(14, 8)	16.284	0
X(7, 2, 2)	0	41.952	X(14, 13, 1)	0	59.156	DIST(14, 9)	97.06	0
X(7, 2, 3)	0	41.952	X(14, 13, 2)	0	59.156	DIST(14, 10)	24.932	0
X(7, 3, 1)	0	26.036	X(14, 13, 3)	0	59.156	DIST(14, 11)	37.812	0
X(7, 3, 2)	0	26.036	X(14, 14, 1)	0	460	DIST(14, 12)	50.968	0
X(7, 3, 3)	0	26.036	X(14, 14, 2)	0	460	DIST(14, 13)	59.156	0
X(7, 4, 1)	0	27.232	X(14, 14, 3)	0	460	DIST(14, 14)	460	0
X(7, 4, 2)	0	27.232	X(14, 15, 1)	0	11.224	DIST(14, 15)	11.224	0
X(7, 4, 3)	0	27.232	X(14, 15, 2)	0	11.224	DIST(15, 1)	67.712	0
X(7, 5, 1)	0	24.472	X(14, 15, 3)	0	11.224	DIST(15, 2)	20.332	0
X(7, 5, 2)	0	24.472	X(15, 1, 1)	0	67.712	DIST(15, 3)	65.964	0
X(7, 5, 3)	0	24.472	X(15, 1, 2)	0	67.712	DIST(15, 4)	34.868	0
X(7, 6, 1)	0	38.916	X(15, 1, 3)	0	67.712	DIST(15, 5)	40.204	0
X(7, 6, 2)	0	38.916	X(15, 2, 1)	0	20.332	DIST(15, 6)	52.44	0
X(7, 6, 3)	0	38.916	X(15, 2, 2)	0	20.332	DIST(15, 7)	62.284	0
X(7, 7, 1)	0	460	X(15, 2, 3)	0	20.332	DIST(15, 8)	5.152	0
X(7, 7, 2)	0	460	X(15, 3, 1)	0	65.964	DIST(15, 9)	85.836	0
X(7, 7, 3)	0	460	X(15, 3, 2)	0	65.964	DIST(15, 10)	14.26	0
X(7, 8, 1)	0	57.132	X(15, 3, 3)	0	65.964	DIST(15, 11)	26.772	0
X(7, 8, 2)	0	57.132	X(15, 4, 1)	0	34.868	DIST(15, 12)	40.572	0
X(7, 8, 3)	0	57.132	X(15, 4, 2)	0	34.868	DIST(15, 13)	47.932	0
X(7, 9, 1)	0	27.048	X(15, 4, 3)	0	34.868	DIST(15, 14)	11.224	0
X(7, 9, 2)	0	27.048	X(15, 5, 1)	0	40.204	DIST(15, 15)	460	0

Row	Slack or Surplus	Dual Price	Row	Slack or Surplus	Dual Price	Row	Slack or Surplus	Dual Price
1	0	0	250	14	0	499	28	0
2	0	-1	251	28	0	500	14	0
3	242.42	-1	252	28	0	501	14	0

4	5.2	0	253	14	0	502	28	0
5	0	0	254	28	0	503	28	0
6	0	0	255	28	0	504	28	0
7	0	0	256	28	0	505	14	0
8	0	0	257	14	0	506	14	0
9	0	0	258	14	0	507	0	0
10	0	0	259	14	0	508	4	0
11	0	0	260	14	0	509	3	0
12	0	0	261	28	0	510	1	0
13	0	0	262	14	0	511	14	0
14	0	0	263	14	0	512	14	0
15	0	0	264	14	0	513	14	0
16	0	0	265	28	0	514	13	0
17	0	0	266	28	0	515	12	0
18	0	0	267	14	0	516	2	0
19	0	0	268	14	0	517	0	0
20	0	0	269	14	0	518	14	0
21	0	0	270	14	0	519	14	0
22	0	0	271	0	0	520	28	0
23	0	0	272	0	0	521	14	0
24	0	0	273	0	0	522	18	0
25	0	0	274	0	0	523	17	0
26	0	0	275	14	0	524	0	0
27	0	0	276	0	0	525	28	0
28	0	0	277	0	0	526	28	0
29	0	0	278	0	0	527	28	0
30	0	0	279	14	0	528	27	0
31	0	0	280	14	0	529	26	0
32	0	0	281	0	0	530	16	0
33	0	0	282	14	0	531	14	0
34	0	0	283	14	0	532	28	0
35	0	0	284	14	0	533	28	0
36	0	0	285	0	0	534	24	0
37	0	0	286	0	0	535	10	0
38	0	0	287	0	0	536	14	0
39	0	0	288	0	0	537	13	0
40	0	0	289	14	0	538	11	0
41	0	0	290	0	0	539	24	0
42	0	0	291	0	0	540	24	0
43	0	0	292	0	0	541	24	0
44	0	0	293	14	0	542	23	0
45	0	0	294	14	0	543	7	0
46	0	0	295	0	0	544	12	0
47	0	0	296	28	0	545	10	0
48	0	0	297	28	0	546	24	0
49	0	0	298	28	0	547	24	0
50	0	0	299	14	0	548	25	0
51	0	0	300	14	0	549	11	0
52	0	0	301	14	0	550	0	0
53	0	0	302	14	0	551	14	0
54	0	0	303	28	0	552	12	0
55	0	0	304	14	0	553	25	0
56	0	0	305	14	0	554	25	0
57	0	0	306	14	0	555	25	0
58	0	0	307	28	0	556	24	0
59	0	0	308	28	0	557	23	0
60	0	0	309	14	0	558	13	0
61	0	0	310	14	0	559	11	0
62	0	0	311	14	0	560	25	0
63	0	0	312	0	0	561	25	0
64	0	0	313	14	0	562	27	0
65	0	0	314	14	0	563	13	0
66	0	0	315	0	0	564	17	0
67	0	0	316	0	0	565	16	0
68	0	0	317	14	0	566	14	0
69	0	0	318	0	0	567	27	0
70	0	0	319	0	0	568	27	0
71	0	0	320	14	0	569	27	0
72	0	0	321	14	0	570	26	0
73	0	0	322	14	0	571	25	0
74	0	0	323	0	0	572	0	0
75	0	0	324	14	0	573	13	0
76	0	0	325	14	0	574	27	0
77	0	0	326	0	0	575	27	0
78	0	0	327	14	0	576	14	0
79	0	0	328	14	0	577	0	0
80	0	0	329	0	0	578	4	0
81	0	0	330	0	0	579	3	0
82	0	0	331	14	0	580	1	0
83	0	0	332	0	0	581	14	0
84	0	0	333	0	0	582	14	0
85	0	0	334	14	0	583	14	0
86	0	0	335	14	0	584	13	0
87	0	0	336	14	0	585	12	0
88	0	0	337	0	0	586	2	0
89	0	0	338	28	0	587	0	0
90	0	0	339	28	0	588	14	0
91	0	0	340	14	0	589	14	0
92	0	0	341	28	0	590	14	0
93	0	0	342	28	0	591	0	0

94	0	0	343	14	0	592	4	0
95	423	0	344	14	0	593	3	0
96	6	0	345	28	0	594	1	0
97	35	0	346	14	0	595	14	0
98	0	0	347	14	0	596	14	0
99	0	0	348	28	0	597	14	0
100	0	0	349	28	0	598	13	0
101	0	0	350	28	0	599	12	0
102	0	0	351	14	0	600	2	0
103	0	0	352	14	0	601	0	0
104	0	0	353	14	0	602	14	0
105	0	0	354	0	0	603	14	0
106	0	0	355	14	0	604	14	0
107	0	0	356	14	0	605	0	0
108	0	0	357	0	0	606	4	0
109	0	0	358	0	0	607	3	0
110	0	0	359	14	0	608	1	0
111	0	0	360	0	0	609	14	0
112	0	0	361	0	0	610	14	0
113	0	0	362	14	0	611	14	0
114	14	0	363	14	0	612	13	0
115	14	0	364	14	0	613	12	0
116	14	0	365	0	0	614	2	0
117	0	0	366	14	0	615	0	0
118	0	0	367	14	0	616	14	0
119	0	0	368	0	0	617	14	0
120	0	0	369	14	0	618	15	0
121	14	0	370	14	0	619	1	0
122	0	0	371	0	0	620	5	0
123	0	0	372	0	0	621	4	0
124	0	0	373	14	0	622	2	0
125	14	0	374	0	0	623	0	0
126	14	0	375	0	0	624	15	0
127	0	0	376	14	0	625	15	0
128	14	0	377	14	0	626	14	0
129	14	0	378	14	0	627	13	0
130	14	0	379	0	0	628	3	0
131	0	0	380	28	0	629	1	0
132	0	0	381	28	0	630	15	0
133	0	0	382	14	0	631	15	0
134	0	0	383	28	0	632	16	0
135	14	0	384	28	0	633	2	0
136	0	0	385	14	0	634	6	0
137	0	0	386	14	0	635	5	0
138	0	0	387	28	0	636	3	0
139	14	0	388	14	0	637	16	0
140	14	0	389	14	0	638	16	0
141	0	0	390	28	0	639	16	0
142	14	0	391	28	0	640	0	0
143	14	0	392	28	0	641	14	0
144	14	0	393	14	0	642	4	0
145	0	0	394	28	0	643	2	0
146	0	0	395	28	0	644	16	0
147	0	0	396	14	0	645	16	0
148	0	0	397	28	0	646	26	0
149	14	0	398	28	0	647	12	0
150	0	0	399	14	0	648	16	0
151	0	0	400	14	0	649	0	0
152	0	0	401	28	0	650	13	0
153	14	0	402	14	0	651	26	0
154	14	0	403	14	0	652	26	0
155	0	0	404	28	0	653	26	0
156	28	0	405	28	0	654	25	0
157	28	0	406	28	0	655	24	0
158	28	0	407	14	0	656	14	0
159	14	0	408	14	0	657	12	0
160	14	0	409	14	0	658	26	0
161	14	0	410	0	0	659	26	0
162	14	0	411	14	0	660	28	0
163	28	0	412	14	0	661	14	0
164	14	0	413	0	0	662	18	0
165	14	0	414	0	0	663	17	0
166	14	0	415	14	0	664	15	0
167	28	0	416	0	0	665	28	0
168	28	0	417	0	0	666	28	0
169	14	0	418	14	0	667	28	0
170	28	0	419	14	0	668	27	0
171	28	0	420	14	0	669	26	0
172	28	0	421	0	0	670	16	0
173	14	0	422	28	0	671	14	0
174	14	0	423	28	0	672	28	0
175	14	0	424	14	0	673	28	0
176	14	0	425	28	0	674	14	0
177	28	0	426	28	0	675	0	0
178	14	0	427	14	0	676	4	0
179	14	0	428	14	0	677	3	0
180	14	0	429	28	0	678	1	0
181	28	0	430	14	0	679	14	0
182	28	0	431	14	0	680	14	0
183	14	0	432	28	0	681	14	0

184	28	0	433	28	0	682	13	0
185	28	0	434	28	0	683	12	0
186	28	0	435	14	0	684	2	0
187	14	0	436	28	0	685	0	0
188	14	0	437	28	0	686	14	0
189	14	0	438	14	0	687	14	0
190	14	0	439	28	0	688	14	0
191	28	0	440	28	0	689	0	0
192	14	0	441	14	0	690	4	0
193	14	0	442	14	0	691	3	0
194	14	0	443	28	0	692	1	0
195	28	0	444	14	0	693	14	0
196	28	0	445	14	0	694	14	0
197	14	0	446	28	0	695	14	0
198	28	0	447	28	0	696	13	0
199	28	0	448	28	0	697	12	0
200	28	0	449	14	0	698	2	0
201	14	0	450	14	0	699	0	0
202	14	0	451	14	0	700	14	0
203	14	0	452	0	0	701	14	0
204	14	0	453	14	0	702	0	0
205	28	0	454	14	0	703	0	0
206	14	0	455	0	0	704	0	0
207	14	0	456	0	0	705	0	0
208	14	0	457	14	0	706	0	0
209	28	0	458	0	0	707	0	0
210	28	0	459	0	0	708	0	0
211	14	0	460	14	0	709	0	0
212	14	0	461	14	0	710	14	0
213	14	0	462	14	0	711	0	0
214	14	0	463	0	0	712	14	0
215	0	0	464	14	0	713	10	0
216	0	0	465	14	0	714	14	0
217	0	0	466	0	0	715	0	0
218	0	0	467	14	0	716	11	0
219	14	0	468	14	0	717	14	0
220	0	0	469	0	0	718	0	0
221	0	0	470	0	0	719	13	0
222	0	0	471	14	0	720	14	0
223	14	0	472	0	0	721	14	0
224	14	0	473	0	0	722	0	0
225	0	0	474	14	0	723	14	0
226	28	0	475	14	0	724	14	0
227	28	0	476	14	0	725	0	0
228	28	0	477	0	0	726	0	0
229	14	0	478	14	0	727	0	0
230	14	0	479	14	0	728	0	0
231	14	0	480	0	0	729	14	0
232	14	0	481	14	0	730	14	0
233	28	0	482	14	0	731	1	0
234	14	0	483	0	0	732	14	0
235	14	0	484	0	0	733	14	0
236	14	0	485	14	0	734	2	0
237	28	0	486	0	0	735	14	0
238	28	0	487	0	0	736	0	0
239	14	0	488	14	0	737	12	0
240	28	0	489	14	0	738	0	0
241	28	0	490	14	0	739	0	0
242	28	0	491	0	0	740	14	0
243	14	0	492	28	0	741	0	0
244	14	0	493	28	0	742	0	0
245	14	0	494	14	0	743	0	0
246	14	0	495	28	0	744	14	0
247	28	0	496	28	0	745	14	0
248	14	0	497	14	0	746	0	0
249	14	0	498	14	0			