ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΑΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΔΙΑΔΙΚΑΣΙΩΝ ΕΚΚΕΝΩΣΗΣ ΚΤΙΡΙΩΝ ΣΕ ΠΕΡΙΠΤΩΣΗ ΕΚΤΑΚΤΩΝ ΑΝΑΓΚΩΝ

Εκπονήθηκε από τον φοιτητή:

ΣΤΕΛΙΟ ΑΓΑΘΟΚΛΗ

Επιβλέπων Καθηγητής:

Δρ. ΑΘΑΝΑΣΙΟΣ ΖΗΛΙΑΣΚΟΠΟΥΛΟΣ

Υπεβλήθη για την εκπλήρωση μέρους των
απαιτήσεων για την απόκτηση του
Διπλώματος Μηχανολόγου Μηχανικού Βιομηχανίας
2008

ΕΓΚΡΙΘΗΚΕ ΑΠΟ ΤΑ ΜΕΛΗ ΤΗΣ ΤΡΙΜΕΛΟΥΣ ΕΞΕΤΑΣΤΙΚΗΣ ΕΠΙΤΡΟΠΗΣ:

Πρώτος Εξεταστής: Δρ. Αθανάσιος Ζηλιασκόπουλος
(Επιβλέπων) Αναπληρωτής Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών Βιομηχανίας, Πανεπιστήμιο Θεσσαλίας

Δεύτερος Εξεταστής: Δρ Καραμάνος Σπύρος
Επίκουρος Κθηγητής, Τμήμα Μηχανολόγων Μηχανικών Βιομηχανίας, Πανεπιστήμιο Θεσσαλίας

Τρίτος Εξεταστής : Δρ. Γεώργιος Κοζανίδης
Λέκτορας, Τμήμα Μηχανολόγων Μηχανικών Βιομηχανίας, Πανεπιστήμιο Θεσσαλίας

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΧΡΟΝΟΥ ΕΚΚΕΝΩΣΗΣ ΚΤΙΡΙΩΝ ΣΕ ΠΕΡΙΠΤΩΣΗ ΕΚΤΑΚΤΩΝ ΑΝΑΓΚΩΝ

ΠΕΡΙΛΗΨΗ

Σκοπός της εργασίας αυτής είναι η όσο το δυνατό πιο λειτουργική ανάθεση καθηκόντων στο ανθρώπινο δυναμικό μιας εταιρείας, (στη συγκεκριμένη περίπτωση στους μοναχούς ενός μοναστηριού του Αγίου Όρους), έτσι ώστε σε περίπτωση αναγκαστικής εκκένωσης του κτιρίου (σε περίπτωση έκτακτης ανάγκης) να προλάβουμε να περισώσουμε όσο το δυνατό μεγαλύτερο αριθμό αντικειμένων (κειμηλίων στην περίπτωσή μας).

Αρχικά διατυπώνουμε το μαθηματικό πρότυπο το οποίο χρησιμοποιούμε στο πρόβλημά μας.

Ακολούθως με την βοήθεια γλώσσας μαθηματικού προγραμματισμού επιλύουμε το πρόβλημα και αναλύουμε τα αποτελέσματα.

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ θερμά όλους όσοι άμεσα και έμμεσα με βοήθησαν να ολοκληρώσω αυτή τη διπλωματική εργασία:

Κατ' αρχή θα ήθελα να ευχαριστήσω τον Αναπληρωτή Καθηγητή κ. Αθανάσιο Ζηλιασκόπουλο για την ανάθεση και επίβλεψη της εργασίας αυτής.

Επίσης θα ήθελα να ευχαριστήσω τους Αθανάσιο Λόη και Βαγγέλη Κατσαρό που ως συνεπιβλέπων, μου έδωσαν πολύτιμες συμβουλές και κατευθύνσεις για την ολοκλήρωση της προσπάθειας μου.

Τέλος θα ήθελα να ευχαριστήσω τους φίλους και συμφοιτητές μου που βαδίσαμε μαζί όλα αυτά τα χρόνια, και κυρίως τους γονείς μου για την αμέριστη τους συμπαράσταση –ηθική και μη- καθ' όλη τη διάρκεια των προπτυχιακών μου σπουδών.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1: ΠΡΟΛΟΓΟΣ 7

ΚΕΦΑΛΑΙΟ 2: ΑΝΑΣΚΟΠΗΣΗ ΚΑΙ ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΠΡΟΒΛΗΜΑΤΟΣ 10

2.1 Κίνητρο και Υπόβαθρο....................11

2.2 Βιβλιογραφική ανασκόπηση....................13

2.3 Ιστορική ανασκόπηση και περιγραφή του προβλήματος....................15

2.4 Οργάνωση της διπλωματικής εργασίας....................20

2.5 Συμπεράσματα....................21

ΚΕΦΑΛΑΙΟ 3: ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΠΡΟΒΛΗΜΑΤΟΣ....................22

3.1 Μεθοδολογία....................23

3.2 Μαθηματικό Μοντέλο....................28

3.3 Επιλογή Αλγορίθμου 32

3.4 Συμπεράσματα....................34

ΚΕΦΑΛΑΙΟ 4: ΔΕΔΟΜΕΝΑ –ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ....................34

4.1 Ένας μοναχός - Τέσσερα αντικείμενα....................36

4.1.1 Απλός ευρετικός αλγόριθμος....................36

4.1.2 AMPL (CPLEX)....................38

4.1.3 Συμπεράσματα και ανάλυση αποτελεσμάτων....................39

4.2 Ένας μοναχός - Πέντε αντικείμενα....................41

4.2.1 Συμπεράσματα και ανάλυση αποτελεσμάτων....................42

4.3 Τρεις μοναχοί – Δέκα αντικείμενα....................43

4.3.1 Συμπεράσματα και ανάλυση πειράματος....................44

ΚΕΦΑΛΑΙΟ 5: ΤΕΛΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ....................46

ΒΙΒΛΙΟΓΡΑΦΙΑ....................49

ΠΑΡΑΡΤΗΜΑ [Α]......52

ΠΑΡΑΡΤΗΜΑ [Β]......55

ΠΑΡΑΡΤΗΜΑ [Γ]......57

ΠΑΡΑΡΤΗΜΑ [Δ]......60

ΠΑΡΑΡΤΗΜΑ [Ε]......70

ΠΑΡΑΡΤΗΜΑ [Ζ]......74

ΚΕΦΑΛΑΙΟ 1
ΠΡΟΛΟΓΟΣ

ΚΕΦΑΛΑΙΟ 1: ΠΡΟΛΟΓΟΣ

Κατά την εκκένωση ενός κτιρίου πρώτιστο μέλημά μας είναι η ασφάλεια του προσωπικού που βρίσκεται στο υπό εκκένωση κτίριο. Ακολούθως αφού έχουμε διασφαλίσει την ασφάλεια των ανθρώπων προσπαθούμε με συντονισμένη προσπάθεια να περισώσουμε όσο το δυνατό περισσότερα αντικείμενα τα οποία θεωρούμε υψηλής αξίας.

Τον συντονισμό αυτό θα προσπαθήσουμε να επιτύχουμε στα πλαίσια αυτής της διπλωματικής εργασίας.

Στο συγκεκριμένο πρόβλημα που μας ανατέθηκε έχουμε ένα μοναστήρι στο οποίο όπως είναι φυσικό υπάρχουν ιερά κειμήλια ανεκτίμητης αξίας που φυλάσσονται εκεί για αιώνες. Αυτό που εμείς καλούμαστε να κάνουμε είναι να προστατεύσουμε όσο το δυνατό περισσότερα αντικείμενα, σε περίπτωση κάποιας έκτακτης ανάγκης για εκκένωση της μονής, δίνοντας έμφαση στα αντικείμενα που δεν αντικαθίστανται και φυσικά όχι στον αριθμό που θα περισώσουμε.

Ζητούμενο μας δηλαδή σε αυτό το πρόβλημα είναι ότι τα αντικείμενα θα πρέπει να μεταφερθούν με ασφάλεια έξω από τους κτιριακούς χώρους για προστασία τους. Για να γίνει αυτό εφικτό πρέπει το ανθρώπινο προσωπικό που δραστηριοποιείται σε αυτούς τους χώρους να επωμιστεί την μεταφορά των αντικειμένων αυτών. Η παραπάνω διαδικασία θα πρέπει να γίνει στα πλαίσια ενός αυστηρού χρονοδιαγράμματος γι'αυτό εμείς θα πρέπει να φτιάξουμε ένα μαθηματικό πρότυπο σύμφωνα με το οποίο θα ελαχιστοποιούμε τον χρόνο εκκένωσης των κειμηλίων.

Η ανάπτυξη του μαθηματικού μοντέλου ή και του αλγορίθμου που καλούμαστε να εφαρμόσουμε περιλαμβάνει:

α)Έρευνα η οποία συγκεντρώνει παλαιότερα μοντέλα (αν υπάρχουν) για παρόμοια προβλήματα.

β) **Αξιολόγηση** των τεχνικών επίλυσης οι οποίες πρέπει να ικανοποιούν τους περιορισμούς που προκύπτουν από το πρόβλημα.

Στο πρόβλημα που εμείς καλούμαστε να επιλύσουμε υπάρχουν και μερικοί περιορισμοί οι οποίοι θα πρέπει να ικανοποιούνται μέσα από το μαθηματικό μοντέλο που θα εφαρμόσουμε:

α)Το μέγεθος και το βάρος των αντικειμένων είναι ένα ζήτημα το οποίο θα μας αναγκάσει να το λάβουμε υπόψη μας στην εφαρμογή του μοντέλου. Υπάρχουν αντικείμενα που έχουν μεγάλο μέγεθος και έτσι για την μεταφορά τους χρειάζεται περισσότερο από ένας μοναχός.

β)Η ικανότητα μεταφοράς που έχει ο κάθε μοναχός είναι ακόμα ένας περιορισμός αφού δεν μπορεί να μεταφέρει ταυτόχρονα περισσότερα από ένα συγκεκριμένο αριθμό αντικειμένων σύμφωνα με το βάρος ή και τον όγκο.

γ)Ίσως ο σημαντικότερος περιορισμός είναι αυτός του χρόνου αφού για να γίνει η πιο πάνω διαδικασία εκκένωσης δεν έχουμε στη διάθεσή μας άπειρο χρόνο. Ο χρόνος είναι αυτός που μας πιέζει και θα πρέπει να περισώσουμε όσο το δυνατό περισσότερα αντικείμενα. Γι'αυτό θα πρέπει να ξέρουμε πόσος χρόνος χρειάζεται για να μετακινηθούν οι μοναχοί από το ένα σημείο στο άλλο μαζεύοντας τα κειμήλια καθώς επίσης και πόσος χρόνος τους χρειάζεται ώστε να τα μεταφέρουν σε ασφαλέστερο περιβάλλον.

ΚΕΦΑΛΑΙΟ 2
ΑΝΑΣΚΟΠΗΣΗ ΚΑΙ ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΠΡΟΜΛΗΜΑΤΟΣ

ΚΕΦΑΛΑΙΟ 2: ΑΝΑΣΚΟΠΗΣΗ ΚΑΙ ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΠΡΟΒΛΗΜΑΤΟΣ

2.1 ΚΙΝΗΤΡΟ ΚΑΙ ΥΠΟΒΑΘΡΟ

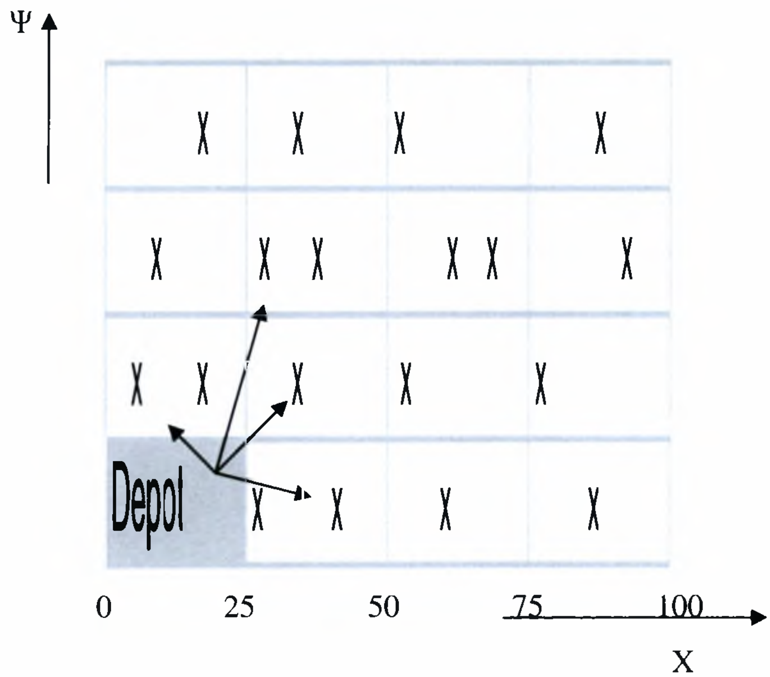
Τα τελευταία χρόνια όλο και περισσότεροι επιστήμονες ασχολούνται με τα προβλήματα δρομολόγησης οχημάτων (vehicle routing problem) και αυτό λόγω των ταχύτερων ρυθμών ανάπτυξης της παγκόσμιας οικονομίας, μολονότι για να επιβιώσει σήμερα μια επιχείρηση μεταφοράς αγαθών, πρέπει να κινείται σε ευέλικτους ρυθμούς στην αγορά αφού ο χρόνος παραγωγής του τελικού προϊόντος από τις πρώτες ύλες έχει ελαχιστοποιηθεί.

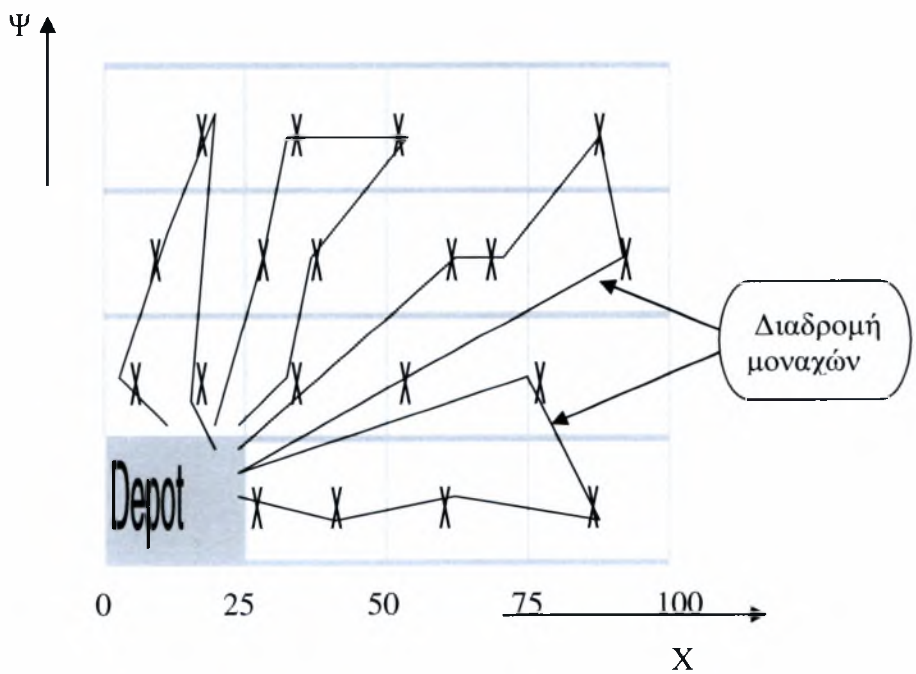
Τα πιο βασικά χαρακτηριστικά τα οποία ξεχωρίζουν τα προβλήματα δρομολόγησης (Bodin, 1981) είναι: η θέση των σημείων από τα οποία θα περάσει η διαδρομή, οι ιδιότητες του οχήματος το οποίο θα δρομολογηθεί και το κριτήριο το οποίο καθορίζει αν μια διαδρομή είναι καλή ή όχι. Τα χαρακτηριστικά αυτά επηρεάζουν τα σχετικά δεδομένα που χρειάζονται και τον βαθμό δυσκολίας της εύρεσης μιας καλής διαδρομής.

Όπως προαναφέραμε αυτή η κατηγορία προβλημάτων έχει ως τελικό σκοπό την εύρεση της ελάχιστης διαδρομής που πρέπει να διανύσουν τα οχήματα μας (στην προκειμένη περίπτωση μοναχοί) εκκινώντας από ένα αρχικό σημείο παραλαμβάνοντας και παραδίδοντας τον μέγιστο αριθμό αντικειμένων σε ένα τελικό σημείο. Ο μέγιστος αριθμός αντικειμένων μεταφράζεται σε μέγιστη αξία αντικειμένων. Το παραπάνω πρόβλημα θα ήταν απλό αν δεν υπήρχαν κάποιοι περιορισμοί που πρέπει κατά την επίλυση του προβλήματος να καλύπτονται.

Το κίνητρο που έχουμε αναλαμβάνοντας αυτή την εργασία είναι ότι για το συγκεκριμένο πρόβλημα, δηλαδή της εκκένωσης πολύτιμων αντικειμένων από τους ανθρώπους, στην περίπτωση κάποιας έκτακτης ανάγκης, υπάρχει έλλειψη μοντέλων αφού όπως είναι λογικό σε τέτοιες περιπτώσεις προτεραιότητα έχει ο άνθρωπος. Αποτέλεσμα της παραπάνω παραδοχής είναι ότι υπόβαθρο πάνω στο οποίο θα στηριχτούμε για την εργασία μας δεν υπάρχει αυτή τη στιγμή, γι' αυτό θα χρησιμοποιηθούν καινούργιες μέθοδοι του προβλήματος **Pick up and delivery** .

Τα προβλήματα **Pick up and delivery** περιλαμβάνονται στην κατηγορία VRP (Vehicle Routing Problem), τα οποία περιλαμβάνουν ένα σχέδιο ελαχίστων διαδρομών του συνόλου ενός στόλου οχημάτων, που έχουν αρχικό και τελικό σημείο αποθήκης και η διαδρομή των οποίων είναι γνωστή εκ των προτέρων. Στην περίπτωσή μας εμείς θα έχουμε αντί για στόλο οχημάτων άγημα μοναχών και η διαδρομή που θα πρέπει να ακολουθήσει ο κάθε μοναχός θα την βρούμε σύμφωνα με την αξία τον αντικειμένων που βρίσκονται μέσα στο χώρο που θα θέλουμε να εκκενώσουμε και την θέση του κάθε αντικειμένου.





2.2 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

Καθώς η τεχνολογία αναπτύσσεται, όλο και μεγαλύτερη είναι η ανάγκη κατανόησης και εφαρμογής βέλτιστης λύσης στα προβλήματα απόφασης σε πραγματικό χρόνο. Η αποδοτική διανομή των αγαθών συνεπάγεται, μεταξύ άλλων, έναν προσδιορισμό των διαδρομών και των προγραμμάτων για το στόλο των οχημάτων έτσι ώστε οι συνολικές δαπάνες διανομής να ελαχιστοποιούνται, ενώ οι διάφορες απαιτήσεις (περιορισμοί) καλύπτονται. Οι περιορισμοί αφορούν τις ικανότητες οχημάτων, χρονικά παράθυρα καθώς επίσης και κάποιες ιδιαιτερότητες που μπορεί να έχουν συγκεκριμένα αντικείμενα. Όλα τα οχήματα ξεκινούν από ένα συγκεκριμένο σημείο εκκίνησης και αφού συμπληρώσουν την διαδρομή τους καταλήγουν και πάλι σε ένα προκαθορισμένο σημείο από όπου θα έχουν την δυνατότητα να ξεκινήσουν μία καινούργια διαδρομή εάν βέβαια έχουν τα χρονικά περιθώρια. Για κάθε αντικείμενο τώρα έχουμε και πάλι ένα συγκεκριμένο σημείο παραλαβής και παράδοσης και για κάθε ένα από αυτά δίνεται ένα χρονικό παράθυρο. Για τα σημεία παραλαβής το χρονικό περιθώριο αυτό είναι το χρονικό διάστημα μέχρι να γίνει η παραλαβή του αντικειμένου από τον μοναχό ενώ το χρονικό παράθυρο για τα σημεία παράδοσης σημαίνει ότι μέσα στο χρονικό περιθώριο αυτό θα πρέπει να γίνει η παράδοση του αντικειμένου. Ένας άλλος περιορισμός είναι η ικανότητα μεταφοράς που έχουν οι μοναχοί, άρα δεν μπορούμε από άνα μοναχό, να απαιτήσουμε να μεταφέρει περισσότερα αντικείμενα από όσα ενδείκνυται ότι μπορεί να μεταφέρει. Ο τελικός σκοπός λοιπόν είναι η ελαχιστοποίηση της διαδρομής που θα καλύψουν τα οχήματά μας ενώ παράλληλα έχουμε μεγιστοποίηση του συνολικού κόστους των αντικειμένων που μεταφέραμε, δεδομένου της ικανοποίησης των χρονικών περιορισμών και όχι μόνο.

Ένα μεγάλο μέρος της έρευνας στον τομέα του logistics management έχει επικεντρωθεί στην ανάπτυξη αποτελεσματικών αλγορίθμων για τα διάφορα προβλήματα βελτιστοποίησης διαδρομών. Ως εκ τούτου η βιβλιογραφία για τα προβλήματα DARP είναι εκτενής. Εν γένει, οι αλγόριθμοι που έχουν κατασκευαστεί για το πρόβλημα DARP μπορούν να κατηγοριοποιηθούν ως exact heuristics και ως υβριδικοί.

Ο Χ. Ψαραύτης [1] έχει περιγράψει μία ακριβή προσέγγιση σε DARPTW βασισμένη στο δυναμικό προγραμματισμό. Αργότερα και πάλι ο ίδιος, Ψαραύτης [2], πρότεινε ένα ποιο εκτεταμένο αλγόριθμο δυναμικού προγραμματισμού για τη στατική περίπτωση του προβλήματος και τον επέκτεινε στην δυναμική περίπτωση. Η αντικειμενική συνάρτηση εδώ σχηματίστηκε ως άθροισμα του συνολικού χρόνου που χρειάζεται για να πραγματοποιηθεί η διαδρομή και της συνολικής δυσαρέσκειας των πελατών. Τελικά ο ίδιος πρότεινε μία παραλλαγή του ίδιου αλγόριθμου για την περίπτωση που χρονικά περιθώρια καθορίζονται για κάθε σταθμό και ο συνολικός χρόνος διαδρομής ελαχιστοποιείται στην αντικειμενική συνάρτηση. Οι Jaw et al. [5] παρουσίασαν παράλληλα ένα ευρετικό αλγόριθμο για το DARPTW και ανέπτυξαν μια ευρετική μέθοδο εισαγωγής για την περίπτωση του στατικού προβλήματος με πολλά οχήματα και χρονικούς περιορισμούς. Οι Desrosiers et al [4] κατασκεύασαν ένα δυναμικό αλγόριθμο για την περίπτωση ενός οχήματος. Στην περίπτωση όμως αυτή η δημιουργία του προβλήματος εμπεριείχε περιορισμούς χρονικών ορίων όπως επίσης περιορισμούς χωρητικότητας του οχήματος και περιορισμούς προτεραιότητας. Οι Dumas et al. [3] παρουσίασαν έναν ακριβή αλγόριθμο για την περίπτωση πολλών οχημάτων του προβλήματος «pick up and delivery». Ο αλγόριθμος αυτός χρησιμοποιεί μία διάταξη δημιουργίας στηλών με ένα υποπρόγραμμα, το οποίο είναι υπεύθυνο για την εύρεση των μικρότερων διαδρομών με βάση τους περιορισμούς. Οι Sexton και Bodin [6] και [7] πρότειναν μία μικτή ακέραια –μη γραμμική εφαρμογή και έναν αλγόριθμο για την επίλυση της περίπτωσης ενός οχήματος. Έπειτα, χρησιμοποίησαν τον αλγόριθμο αυτό μέσα σε ένα νέο ευρετικό αλγόριθμο, για την περίπτωση πολλών οχημάτων [19], ο οποίος εκκινεί ομαδοποιώντας τους πελάτες (αντικείμενα) που βρίσκονται κοντά γεωγραφικά. Χρησιμοποίησαν τη μέθοδο διάσπασης του Bender και έλυσαν το χρονικό πρόβλημα και το πρόβλημα διαδρομών ξεχωριστά.. Οι Desrosiers et al [15] βελτίωσαν εκ νέου την ιδέα αυτή σχηματίζοντας μικρές ομάδες πελατών (“mini-clusters”), οι οποίες συνδυάζονται με τον βέλτιστο τρόπο χρησιμοποιώντας τεχνικές δημιουργίας στηλών. Οι Ioachim et al. [10] πρότειναν μία ευρετική τεχνική, η οποία χρησιμοποιεί μία τεχνική βελτιστοποίησης, με σκοπό να κατασκευάσει τις καλύτερες δυνατές ομάδες. Οι Boerdorfer et al.[12] ανέπτυξαν ένα ευρετικό αλγόριθμο για το στατικό πρόβλημα DARP πολλών οχημάτων, το οποίο επιλυόταν με τη μέθοδο branch and cut. Τέλος μέθοδοι τοπικής αναζήτησης (local search) προτάθηκαν από τους Brugger et al. [16], Healey και Moll [13], Savelsbergh [11] και Thompson και Psaraftis [14].

Τα τελευταία χρόνια πολλοί μοντέρνοι ευρετικοί και μετά-ευρετικοί αλγόριθμοι έχουν αναπτυχθεί για το πρόβλημα DARP. Οι Toth και Vigo [18], ανέπτυξαν έναν ευρετικό αλγόριθμο παράλληλης εισαγωγής για την δυναμική περίπτωση του προβλήματος DARP πολλών οχημάτων που χρησιμοποιούσε εναλλαγές εντός των διαδρομών και τεχνικές απαγορεύσεων [18]. Άλλα παραδείγματα περιλαμβάνουν δυναμικές τεχνικές προσομοίωσης [16], τεχνικές αναζήτησης tabu [17] και τεχνικές regret εισαγωγής [15].

2.3 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΚΑΙ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Εστιάζοντας τώρα στο πρόβλημά μας βλέπουμε την χρησιμότητα διεκπεραίωσης αυτής της εργασίας καθώς επίσης και εφαρμογής της, αφού βλέπουμε ότι κατά την διάρκεια προηγούμενων ετών αλλά και στο παρόν πολλές φορές ακούμε για καταστροφές στην πολιτιστική μας κληρονομιά, γεγονός που μας κάνει να σκεφτούμε τρόπους αντιμετώπισης του προβλήματος αυτού. Οι πυρκαγιές είναι η κύρια αιτία του προβλήματος αυτού και έχουμε ορισμένα παραδείγματα για αναφορά που καταδεικνύουν την αναγκαιότητα υπεράσπισης των κειμηλίων και των διαφόρων πολύτιμων αντικειμένων που εμπερικλείονται στους χώρους αυτούς. Το μοναστήρι της Φανερωμένης που βρίσκεται δυτικά της πόλης της Λευκάδας το οποίο είναι από τα σπουδαιότερα μνημεία του τόπου και το μοναδικό του νησιού έχει ξαναχτιστεί κατά τον 19ο αιώνα, μετά από δύο πυρκαγιές που το κατέστρεψαν.

- ***Το μοναστήρι της Φανερωμένης***

Η Ιερά μονή των Τεσσαράκοντα Μεγαλομαρτύρων που βρίσκεται πολύ κοντά στην πόλη της Σπάρτης γνώρισε και αυτή μεγάλες δοκιμασίες στο πέρασμα του χρόνου και πολλές φορές απειλήθηκε με ολοκληρωτική καταστροφή. Το 1770 η Μονή πυρπολήθηκε από στίφη Αλβανών, ενώ το 1826 οι Αιγύπτιοι την παρέδωσαν στη φωτιά για δεύτερη φορά. Οι αλλεπάλληλες αυτές πυρκαγιές είχαν ως αποτέλεσμα να καταστραφούν εν μέρει οι υπέροχες αγιογραφίες.

- ***Ιερά μονή των Τεσσαράκοντα Μεγαλομαρτύρων***

Το Άγιο Όρος που αποτελεί το πλέον φημισμένο θρησκευτικό μέρος της εκκλησίας μας δεν θα μπορούσε να μην δεχτεί πλήγματα από τις πυρκαγιές στο πέρασμα του χρόνου. Η Ιερά Μονή Χιλιανδαρίου στις 4 Μαρτίου 2004 παραδόθηκε στις φλόγες. Οι μοναχοί έτρεχαν να σώσουν ότι μπορούν συμβάλλοντας στην προσπάθεια της πυροσβεστικής υπηρεσίας. Είναι αδιανόητο να γίνονται σήμερα τέτοιες καταστροφές και να εξαφανίζεται η πολιτιστική μας κληρονομιά σε μερικά λεπτά. Υπάρχουν τα μέσα να προστατευτούν αυτά τα μνημεία σήμερα και όμως όπως μαθαίνουμε δεν υπήρχε νερό στο σύστημα πυρόσβεσης την στιγμή που εκδηλώθηκε η πυρκαγιά. Τελικός απολογισμός, 4 εκκλησίες με παλιές τοιχογραφίες καταστράφηκαν ολοσχερώς, καθώς επίσης λειψανοθήκες και κειμήλια.

- ***Η στιγμή της καταστροφής***

- ***Το ανυπολόγιστο μέγεθος της καταστροφής***

Κατανοώντας τις καταστροφές που μέσα από το πέρασμα των αιώνων έχουν αφήσει το στίγμα τους σε διάφορα μοναστήρια καθίσταται αναγκαίο πλέον να υπάρχουν συστήματα προστασίας για κάθε περίπτωση. Κατ' αρχάς τοποθετούνται σε διάφορα σημεία του μοναστηριακού χώρου πυροσβεστικά υλικά (βαρέλια με νερό ή και με άμμο, φτυάρια) καθώς επίσης και πυροσβεστήρες σε αρκετά σημεία εντός του μοναστηριού. Ένα δεύτερο μέτρο αντιμετώπισης τέτοιου είδους καταστάσεων είναι η συγκρότηση ομάδας πυροπροστασίας εντός του μοναστηριού που θα απαρτίζεται κατά κύριο λόγω από το μόνιμο προσωπικό του ναού δηλαδή τους μοναχούς. Τα αγήματα αυτά έχουν κατά κύριο λόγω την ευθύνη να σώσουν τα μοναστήρια από την καταστροφή. Μόλις υποπέσει κάτι στην αντίληψή τους που θα μπορούσε να επιφέρει ζημιές στο έμψυχο δυναμικό αλλά και στα πολύτιμα κειμήλια που φυλάσσονται στη μονή τότε άμεσα αντιδρούν για την διάσωσή τους. Με συντονισμένες προσπάθειες ο καθένας στο πόστο του προσπαθούν να θέσουν υπό τον έλεγχό τους την οποιανδήποτε απειλή αν και στην προκειμένη περίπτωση οι πυρκαγιές είναι ο μεγαλύτερος κίνδυνος που ενδέχεται να αντιμετωπίσουν. Μέσα από την εξέλιξη της τεχνολογίας τώρα βλέπουμε κάποια μοναστήρια που έχουν την οικονομική ευχέρεια να επιλέγουν νέες μεθοδολογίες και τρόπους ασφάλειας της Μονής τους. Χρησιμοποιώντας την τεχνολογία των οπτικών ινών εγκαθιστούν στην οροφή της Μονής ή και στους γύρω χώρους συστήματα φωτοηλεκτρονικών και θερμοδιαφορικών ανιχνευτών καπνού τα οποία σε ειδικές περιπτώσεις ενεργοποιούν ένα αυτόματο σύστημα πυρόσβεσης.

- ***Καμένη όλη η γύρω περιοχή από τη Μονή***

Βλέποντας τις πιο πάνω εικόνες καταλαβαίνουμε πόσο επιτακτική είναι η ανάγκη για καθορισμό συντονισμένων ενεργειών σε περίπτωση πυρκαγιάς, με στόχο να μην υποστούν οποιανδήποτε ζημιά τα ιερά κειμήλια που φυλάσσονται στα Μοναστήρια μας και που αποτελούν την πολιτιστική μας κληρονομιά.

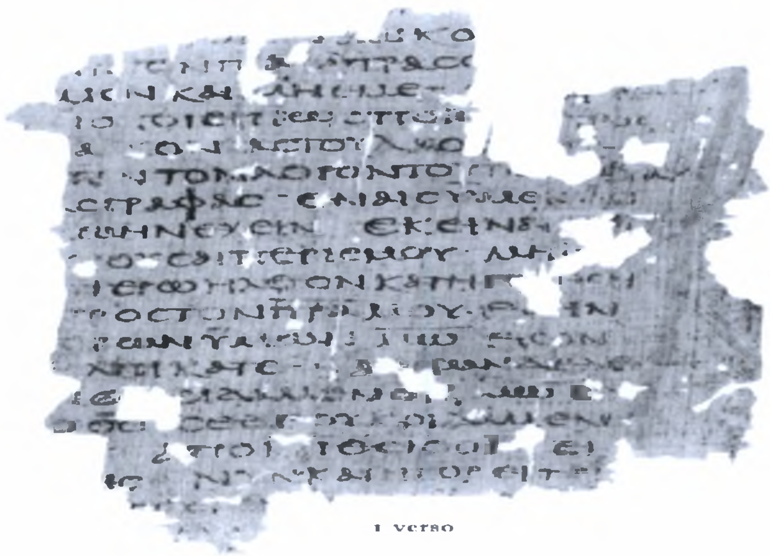
Κάθε εργασία και κάθε πρόβλημα έχει τις δικές του ιδιαιτερότητες που το καθιστούν ξεχωριστό. Έτσι και εμείς σε αυτή την ενότητα θα επισημάνουμε μερικές παραμέτρους που θα μας προβληματίσουν στη συνέχεια της διπλωματικής.

Το μέγεθος των αντικειμένων μπορεί να αναφερθεί ως ένα ιδιαίτερο χαρακτηριστικό του προβλήματος καθότι ένα ιδιαίτερα μεγάλο αντικείμενο, παρουσιάζει δυσκολία στη μετακίνησή του αφού πρόκειται να μεταφερθεί δια μέσου της πόρτας. Σε αυτή τη περίπτωση ο μοναχός θα πρέπει να προσφύγει σε άλλες μεθόδους διάσωσής του ή να εγκαταλείψει την προσπάθεια.

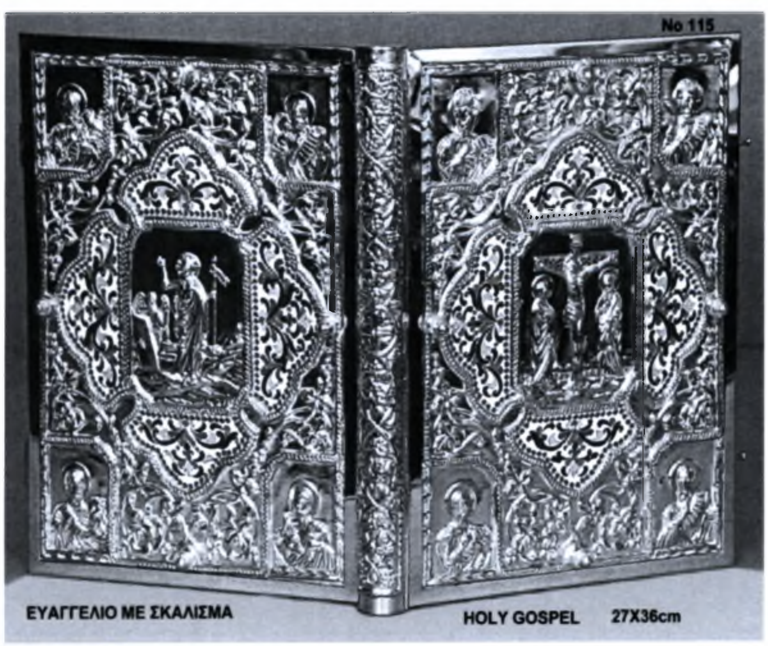
Το βάρος των αντικειμένων επίσης θεωρείται χαρακτηριστικό του προβλήματος διότι υπάρχουν αντικείμενα που χρειάζονται περισσότερους από ένα μοναχούς για να μεταφερθούν.

Το πρόβλημα εμείς το επιλύουμε σε δισδιάστατο επίπεδο. Σε ένα πραγματικό μοναστήρι όμως ενδέχεται τα κειμήλια να υπάρχουν διασκορπισμένα σε διάφορους χώρους της Μονής, καθώς και σε διαφορετικό επίπεδο απ'ότι εμείς επιλύουμε. Μπορεί μερικά αντικείμενα να είναι τοποθετημένα σε ύψος που ένας μοναχός να μην έχει άμεσα πρόσβαση οπότε ο δείκτης δυσκολίας στη διάσωση αυτών αυξάνεται.

Τα παραπάνω χαρακτηριστικά είναι ενδεικτικά της δυσκολίας του προβλήματος που θα αντιμετωπίσουμε. Θα προσπαθήσουμε να τα θέσουμε εντός των περιορισμών του προβλήματος για να πάρουμε όσο το δυνατό πιο ρεαλιστικά αποτελέσματα.



- *Πάπυρος*



- ***Κάλυμμα Ευαγγελίου***

2.4 ΟΡΓΑΝΩΣΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Το υπόλοιπο αυτής της διπλωματικής εργασίας χωρίζεται σε τρεις ενότητες,που καταλαμβάνουν τα κεφάλαια 3 - 5 αντίστοιχα .

Στο Κεφάλαιο 3 γίνεται αναφορά στην μεθοδολογία που ακολουθείται κατά την διαδικασία εκκένωσης ενός κτιρίου από το έμψυχο υλικό που βρίσκεται μέσα και κατά παρόμοιο τρόπο θα γίνει και η εκκένωση των υλικών αγαθών. Ακολούθως επιλέγουμε τον αλγόριθμο που ενδείκνυται να χρησιμοποιήσουμε για την επίλυση του προβλήματος καθώς επίσης μοντελοποιούμε το πρόβλημά μας.

Στο Κεφάλαιο 4 παρατίθενται για τρία παραδείγματα δεδομένα και διαδικασία επίλυσής τους καθώς επίσης και αποτελέσματα επίλυσης. Τα αποτελέσματα προκύπτουν από ένα σύνολο πειραμάτων που σκοπό είχαν να αποδείξουν ότι ο απλός ευρετικός αλγόριθμος που χρησιμοποιήσαμε έχει την ικανότητα να μας δώσει τα θεμιτά αποτελέσματα, άρα είναι λειτουργικός. Επίσης στο κεφάλαιο αυτό πιστοποιείται η λειτουργικότητα του αλγορίθμου μας μέσα από μαθηματικό προγραμματισμό με χρήση της *AMPL*, με την οποία λύνουμε το πρόβλημά μας έχοντας τα ίδια αποτελέσματα με αυτά του αλγορίθμου.

Τέλος, στο Κεφάλαιο 5 συμπερασματικά καταλήγουμε ότι η όλη διαδικασία που εφαρμόσαμε είχε αποτέλεσμα αφού ο αλγόριθμος που χρησιμοποιήσαμε λειτούργησε χωρίς να μας παρουσιάσει κανένα πρόβλημα. Επιπρόσθετα αφήνουμε περιθώρια βελτίωσης του αλγορίθμου αυτού έτσι ώστε να μπορεί να εφαρμόζεται και για τρισδιάστατα προβλήματα.

2.5 ΣΥΜΠΕΡΑΣΜΑΤΑ

Διαβάζοντας τις προηγούμενες παραγράφους κατανοούμε πλήρως την ανάγκη εύρεσης μιας λύσης στο πρόβλημα αυτό, που καθίσταται απειλή για την πολιτιστική μας κληρονομιά. Όπως προαναφέραμε επίσης, δεν έχει γίνει ούτε στο παρελθόν κάποια παρόμοια έρευνα γεγονός που μας επιβάλει να δώσουμε βάση για την εξασφάλιση ορθών αποτελεσμάτων. Σαν μια νέα μέθοδος εύρεσης λύσης παρουσιάζεται αυτή η διπλωματική, αφού κατά το παρελθόν δεν εφαρμόστηκε σε άλλες έρευνες παρόμοιου θέματος ο απλός ευρετικός αλγόριθμος. Εδώ θα προσπαθήσουμε να τον εφαρμόσουμε κάνοντας βέβαια κάποιες μετατροπές στα δεδομένα μας, ώστε να μπορεί να μας δώσει αποτέλεσμα.

ΚΕΦΑΛΑΙΟ 3
ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

ΚΕΦΑΛΑΙΟ 3: ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

3.1 ΜΕΘΟΔΟΛΟΓΙΑ

Κατά την εκκένωση ενός κτιρίου όπως προαναφέραμε πρώτιστο μέλημά μας είναι η εξασφάλιση πλήρους προστασίας και ασφάλειας όλου του έμψυχου υλικού που απαρτίζει την μονάδα και ακολούθως, εφόσον το επιτρέπουν οι συνθήκες η όσο το δυνατό μεγαλύτερη (ποσοτική ή ποιοτική) διάσωση υλικών αγαθών που μας ενδιαφέρουν. Ως εκ τούτου στη βιβλιογραφία που ανατρέξαμε για εκκένωση κτιρίων από διάφορα αντικείμενα (στη προκειμένη περίπτωση ιερά κειμήλια), δεν προέκυψε παρόμοια έρευνα αφού σε τέτοιου είδους εργασίες μας ενδιαφέρει η ασφάλεια των ανθρώπων. Στην παρούσα διπλωματική εργασία θεωρούμε δεδομένη την ασφάλεια των ατόμων που απαρτίζουν το κτίριο και με μεθοδολογία παρόμοια με την δική τους προστασία προσπαθούμε να περισώσουμε όσο περισσότερα κειμήλια μπορούμε από το μοναστήρι που μας ενδιαφέρει.

Όπως και στην περίπτωση των ανθρώπων έτσι και εδώ σε κατάσταση εκτάκτου ανάγκης έχουμε το έμψυχο υλικό (μοναχούς) να συγκεντρώνονται έξω από το κτίριο σε ένα αρχικό σημείο αναφοράς και από εκεί να είναι έτοιμοι να μπουν στο κτίριο εφόσον οι συνθήκες το επιτρέπουν. Ο κάθε μοναχός έχει συγκεκριμένο προορισμό γνωρίζοντας τον εκ των προτέρων, έτσι ώστε να πάρει τα αντικείμενα που του αναλογούν και να βγει έξω. Τα αντικείμενα που οι μοναχοί καλούνται να πάρουν καθορίζονται από εμάς μέσα από τα προγράμματα που θα χρησιμοποιήσουμε στη διπλωματική, σύμφωνα με τις ικανότητες μεταφοράς του κάθε μοναχού, τα χαρακτηριστικά των αντικειμένων (όγκος, βάρος, ευαισθησία) και την θέση των αντικειμένων στον χώρο. Αφού γίνουν οι παραπάνω διαδικασίες τότε οι μοναχοί μεταφέρουν τα κειμήλια σε ένα εξωτερικό χώρο που και αυτός θα είναι προκαθορισμένος, σύμφωνα με την θέση του και την ασφάλεια που θα παρέχει τόσο στους ιδίους όσο και στα κειμήλια.

Τα κειμήλια που καλούμαστε να διασώσουμε θα πρέπει πρώτα να τα διαχωρίσουμε σύμφωνα με την αξία τους αφού εμάς ο σκοπός μας είναι η διασφάλιση μεγαλύτερου κόστους. Η αξία τους αφορά το υλικό που είναι φτιαγμένα (πόσο πολύτιμο είναι ή πόσο ευαίσθητο) και την ηλικία τους. Μερικά από τα κειμήλια θα μπορούσε να είναι διάφορες εικόνες εκατοντάδων χρόνων, διάφορα έργα μικροτεχνίας (βημόθυρα, καλύμματα ευαγγελίων, σταυροί, σφραγίδες), έργα ξυλογλυπτικής καθώς επίσης έγγραφα και αντίγραφα τυπικών εικονογραφημένων χειρογράφων.

Η μεθοδολογία που θα ακολουθήσουμε για να πάρουμε τα θεμιτά αποτελέσματα εν μέρη είναι πλέον γνωστή, και είναι βάση των προβλημάτων ***Pick up and Delivery***. Για να μπορούμε όμως να χρησιμοποιήσουμε τον ευρετικό αλγόριθμο του ***Pick up and Delivery*** που ενδείκνυται να χρησιμοποιήσουμε σε αυτό το πρόβλημα θα πρέπει να μορφοποιήσουμε με τέτοιο τρόπο το πρόβλημά μας ώστε να αναγνωρίζεται από τον αλγόριθμο. Την προαναφερθέντα διαδικασία θα διεκπεραιώσουμε μέσα από τα διάφορα προγράμματα που ακολούθως αναφέρονται εκτενέστερα .

Αρχικά παίρνουμε τα στοιχεία που χρειαζόμαστε από το μοναστήρι (αριθμός κειμηλίων, θέση κειμηλίων, αξία κειμηλίων) και τα βάζουμε σε μορφή πίνακα π.χ:

ΠΙΝΑΚΑΣ 1: Δεδομένα σε μορφή **Matlab**

Κόμβος-Κειμήλιο:	Κόστος-Αξία:	Άξονας-Χ	Άξονας-Ψ
1	0	5	12
2	12	10	36
3	15	20	29
4	20	30	19
5	23	35	36
6	27	25	20
7	30	40	10
8	32	15	20
9	35	20	35
10	40	45	15

Δίνοντας αυτό τον πίνακα στη MATLAB, ακολούθως τρέχουμε το πρόγραμμα που έχουμε φτιάξει ώστε να παίρνουμε όλες τις αποστάσεις μεταξύ των κόμβων, καθώς επίσης και τα κόστη που έχουμε από κόμβο σε κόμβο. Το πρόγραμμα αυτό έγινε σύμφωνα με μια μαθηματική συνάρτηση που έχουμε βγάλει όπου με βάση τις συντεταγμένες Χ και Ψ κάθε κόμβου θα υπολογίζετε η απόσταση μεταξύ τους και μετά γνωρίζοντας πόσος χρόνος χρειάζεται για να διανύσουμε μια συγκεκριμένη απόσταση θα βρίσκουμε τον χρόνο που χρειαζόμαστε από κόμβο σε κόμβο. Ο λόγος που χρειαζόμαστε τους χρόνους από κόμβο σε κόμβο είναι προφανής, για να ξέρουμε αν θα έχουν χρόνο τα οχήματα τα οποία θα βρίσκονται σε διάφορες θέσεις να πάρουν τα υπόλοιπα αντικείμενα.

Για το κόστος τώρα εξαρχής θα πρέπει να αναφέρουμε γιατί το ζητούμε: Αυτό που θέλουμε να μας δίνει ο αλγόριθμός τελικά είναι η min απόσταση που θα διανύσουν τα οχήματά μας. Στην περίπτωσή μας αντί για απόσταση, μας ενδιαφέρει το μέγιστο κόστος που θα συγκεντρώσουν ολοκληρώνοντας τις διαδρομές τους στον επιτρεπτό χρόνο. Άρα εμείς λαμβάνουμε το αντίστροφο του κόστους , δηλαδή παίρνουμε το 1/cost, και ζητούμε το min cost στον χρόνο που έχουμε. Ο τρόπος που βρίσκουμε τα κόστη από κόμβο σε κόμβο γίνεται όπως είπαμε σύμφωνα με την μαθηματική συνάρτηση που θα εφαρμόσουμε και θα εξηγήσουμε παρακάτω. Για παράδειγμά για το κόστος από τον κόμβο 1 στον 2, χρειαζόμαστε το κόστος του τόξου 1 - 2 όπου είναι ίσο με τη διαφορά του κόστους του 1 από το 2 δηλαδή 12. Στο πρόβλημα που εμείς επιλύουμε όμως θέλουμε το min κόστος δηλαδή την min σχέση απόστασης κόστους, γι 'αυτό διαιρούμε με το 1 έτσι το κόστος του arc1-2=1/12. Σε περίπτωση που η διαφορά του κόστους μεταξύ δύο κόμβων βγαίνει αρνητική αυτό σημαίνει ότι η αξία του κόμβου που βρισκόμαστε είναι μεγαλύτερη από τον κόμβο που θέλουμε να πάμε. Εμείς θέλουμε να πάμε στους κόμβους με την μεγαλύτερη αξία, άρα θα πρέπει να κάνουμε θετικό το κόστος αυτού του τόξου και όσα θα προσθέσουμε στο κόστος του τόξου αυτού για να γίνει θετικό, θα πρέπει να τα προσθέσουμε και στα υπόλοιπα τόξα που έχουν σαν αρχή τον συγκεκριμένο κόμβο, ώστε να έχουμε τα θεμιτά αποτελέσματα. Με τη μέθοδο αυτή το κόστος του αρνητικού τόξου γίνεται μονάδα και τα κόστη των υπολοίπων τόξων έχουν το κόστος που θα έπρεπε να έχουν, αφού εμάς μας ενδιαφέρει να δείξουμε στον αλγόριθμό μας ότι πιο κοντά του βρίσκονται οι κόμβοι με το μεγαλύτερο κόστος (στην περίπτωσή μας αφού παίρνουμε το αντίστροφο του κόστους, θέλουμε τους κόμβους με το μικρότερο). Για να γίνει τώρα το αρνητικό τόξο θετικό, προσθέτω το αρνητικό +1 στο κόστος του τόξου αλλά και σε όλα τα υπόλοιπα τόξα που για αρχικό κόμβο έχουν τον συγκεκριμένο κόμβο. Στην περίπτωση του κόστους μεταξύ κόμβου 3 στον κόμβο 2 είναι 1/12-15=1/(-3)=αρνητικό. Εμείς τώρα για να μην είναι αρνητικό προσθέτουμε στον παρονομαστή +4 έτσι ώστε το κόστος σε αυτή την περίπτωση να είναι μονάδα. Ακολούθως για το arc3-4 το κόστος θα είναι 1/(20-15)+4= 1/9 και αυτό γίνεται και στα υπόλοιπα τόξα που ξεκινούν από τον κόμβο 3. Εφαρμόζοντας την προαναφερθέντα διαδικασία γνωρίζουμε τα τόξα με το μεγαλύτερο κόστος έτσι μπορούμε να περάσουμε στο επόμενο βήμα, που είναι η μορφοποίηση των δεδομένων μας ώστε να διαβάζονται από τον αλγόριθμο. Το πρόγραμμα στη MATLAB που δημιουργήσαμε για να επιλύει την παραπάνω διαδικασία παραθέτεται στο παράρτημα της εργασίας.

Αφού τρέξουμε το παραπάνω πρόγραμμα παίρνουμε τα αποτελέσματα σε μορφή πίνακα, όπως φαίνεται παρακάτω:

ΠΙΝΑΚΑΣ 2 : Αποτελέσματα **Matlab (calc)**

Από:	Προς:	Χρόνος:	1/Κόστος	Κόστος:
1	2	24.515	0.083333	12
1	3	22.672	0.066667	15
1	4	25.962	0.05	20
1	5	38.419	0.043478	23
1	6	21.541	0.037037	27
1	7	35.057	0.033333	30
1	8	12.806	0.03125	32
1	9	27.459	0.028571	35
1	10	40.112	0.025	40
2	1	24.515	0.083333	12
2	3	12.207	0.25	4
2	4	26.249	0.11111	9
2	5	25	0.083333	12
2	6	21.932	0.0625	16
2	7	39.699	0.052632	19
.	.	.	.	.
.	.	.	.	.

Στη μορφή που λαμβάνουμε τα παραπάνω αποτελέσματα ο αλγόριθμος που θα χρησιμοποιήσουμε και πάλι δεν μπορεί να τα αναγνωρίσει έτσι είμαστε αναγκασμένοι να τα φέρουμε σε τέτοια μορφή ώστε να γίνει αυτό εφικτό. Για να γίνει αυτό κατορθωτό χρησιμοποιούμε ένα μικρό κώδικα μακροεντολών του ***EXCEL***.

Ο κώδικας μακροεντολών που έχουμε δημιουργήσει για να πάρουμε τα αποτελέσματα στη μορφή που θέλουμε παρουσιάζεται στο παράρτημα. Τα αποτελέσματα που παίρνουμε εμπεριέχουν μέσα όλα τα δεδομένα του προβλήματος όπως τις ικανότητες και τους χρόνους κίνησης των μοναχών, τις αξίες και τα κόστη των τόξων των κόμβων καθώς επίσης και τους περιορισμούς των αντικειμένων.

Όσο αφορά τώρα τον αλγόριθμο που θα χρησιμοποιήσουμε αναφερόμαστε παρακάτω:Υπάρχουν οι απόλυτοι αλγόριθμοι, οι ημιευρετικοί αλγόριθμοι και οι ευρετικοί. Οι απόλυτοι αλγόριθμοι πάσχουν σε σχέση με τα άλλα δύο είδη διότι δεν μπορούν να λύσουν προβλήματα μεγάλου μεγέθους. Εμείς το πρόβλημα που μας έχει ανατεθεί έχει μεγάλο αριθμό αντικειμένων γι' αυτό θα χρησιμοποιήσουμε τον ευρετικό αλγόριθμο που παρουσιάζεται παρακάτω:

3.2 ΜΑΘΗΜΑΤΙΚΟ ΜΟΝΤΕΛΟ

Το χαρακτηριστικό πρόβλημα DARP ορίζεται ως ακολούθως: Λαμβάνοντας υπόψη ένα σύνολο οχημάτων και ένα σύνολο διαδρομών που πρέπει να γίνουν υπολογίστε τη βέλτιστη (για τους επιβάτες ή/και το χειριστή) περιήγηση για κάθε μοναχό, χωρίς παραβίαση ορισμένων περιορισμών. Για κάθε μοναχό, καθορίζεται χαρακτηριστικά μια θέση αναφοράς ("έναρξη" και "τέλος" που μπορούν να μην είναι απαραιτήτως οι ίδιες), μια ικανότητα μεταφοράς και ένας μέγιστος χρόνος κίνησης οχήματος. Οι απαιτήσεις της διαδρομής χαρακτηρίζονται από παραλαβές και παραδόσεις αντικειμένων σε διάφορες θέσεις, χρονικά παράθυρα για την παραλαβή και την παράδοση και μέγιστος χρόνος ολοκλήρωσης της διαδρομής. Αυτά είναι τα αρχικά χαρακτηριστικά γνωρίσματα για τις απαιτήσεις μιας διαδρομής. Περισσότερα χαρακτηριστικά γνωρίσματα μπορούν να ληφθούν υπόψη, όπως "οι συγκεκριμένοι περιορισμοί πελατών", δηλαδή για τα πρόσωπα με ειδικές ανάγκες. Στην τρέχουσα μελέτη μένουμε στα αρχικά χαρακτηριστικά γνωρίσματα.

Τα βασικά χαρακτηριστικά σχετικά με τα οχήματα και τις απαιτήσεις του ταξιδιού παρουσιάζονται στον πίνακα 1:

ΠΙΝΑΚΑΣ 3: Ορισμός μεταβλητών

V = {1,2,3, ,|V|} = Σύνολο μοναχών

$\mathbf{Q_v = \{ Q_1 .. Q_{|V|} \}}$= Η συνολική ικανότητα μεταφοράς όλων των μοναχών *v* στο *V*

$\mathbf{MVRT_V = \{ MVRT_1 .. MVRT_{|V|} \}}$ = Μέγιστος χρόνος διαδρομής όλων των μοναχών v στο V.

N = Ο αριθμός των απαιτούμενων αντικειμένων που πρέπει να μεταφερθούν. Κάθε αντικείμενο επιβάλλει ένα δεδομένο σημείο παραλαβής και παράδοσης.

VSD = {1, 2, … , |V| } = Σημείο αρχικής αναφοράς για κάθε μοναχό.

VED = { 2N +|V|+1, 2N+|V| +2, … ,2N +2|V| } = Τελικό σημείο αναφοράς για κάθε μοναχό.

VSED = { 1, 2, … , |V|, 2N +|V|+1, 2N +|V|+2, … , 2N +2|V| }

$\mathbf{P^+}$ **={|V|+1, |V|+2,… |V|+N }** = Ο αριθμός των κομβικών σημείων παραλαβής των αντικειμένων στο δίκτυο.

P $^{-}$ **={ |V|+N +1, |V|+N +2,... |V|+2N}** = Ο αριθμός των κομβικών σημείων παράδοσης των αντικειμένων στο δίκτυο.

P = P^{+} U P^{-} = Ο συνολικός αριθμός των κομβικών σημείων παραλαβής και παράδοσης των αντικειμένων στο δίκτυο.

P ^{all} **= P U VED U VSD** =Ο συνολικός αριθμός κομβικών σημείων (παραλαβής, παράδοσης, αρχικών και τελικών σημείων αναφοράς μοναχών) στο δίκτυο.

P^{ved} **= P U VED** = Ο συνολικός αριθμός κόμβων παραλαβής, παράδοσης, συν το τελικό σημείο αναφοράς.

P^{vsd} **= P U VSD** = Ο συνολικός αριθμός κόμβων παραλαβής, παράδοσης, συν το αρχικό σημείο αναφοράς.

[e_i l_t] = Χρονικό περιθώριο [νωρίτερα, αργότερα] για κάθε κομβικό σημείο (παραλαβής, παράδοσης, αρχικό –τελικό σημείο αναφοράς) $i \in$ **P** ^{all} .

DRT_i = { DRT_1, .. DRT_N}= Καθορισμός του μέγιστου χρόνου για την κάθε διαδρομή, σε κάθε δεδομένη στιγμή, $i \in P^{+}$

τ_{ij} = χρόνος διαδρομής από τον κόμβο i στο j όπου i,j $\in$ **$P_t{}^{all}$** ,

c_{ij} = κόστος διαδρομής από τον κόμβο i στο j όπου $i,j \in$ **P** ^{all}

s_i = χρόνος παραλαβής ή παράδοσης στον κόμβο $i \in$ **P** ^{all}

d_i = Βάρος αντικειμένου στον κόμβο (παραλαβής ή παράδοσης) $i \in P^{all}$. (+1 για παραλαβές, -1 για παραδόσεις, 0 για το αρχικό – τελικό σημείο αναφοράς)

$x_{ij}^{v}, i, j \in P^{all}, i \neq j$ = 1, όταν η ith μεταφορά (στην χρονική στιγμή *t)* εκτελείται από ένα μοναχό *v* , και 0 σε οποιανδήποτε άλλη περίπτωση..

$T_i^{v}, i \in P^{all}$ = Η χρονική μεταβλητή η οποία περιγράφει την χρονική στιγμή στην οποία αρχίζει η μεταφορά από τον κόμβο *i*.

$L_i^{v}, i \in P^{all}$ = Η μεταβλητή του βάρους η οποία περιγράφει το βάρος που μεταφέρει ο μοναχός σε κάθε κόμβο

Μια συνήθης αντικειμενική συνάρτηση που ελαχιστοποιεί το συνολικό κόστος της διαδρομής είναι η ακόλουθη:

$$\sum_{v \in V} \sum_{i \in P^{all}} \sum_{j \in P^{all}} x_{ij}^{v} c_{ij} .$$

Με τους παρακάτω περιορισμούς:

1. Χρονικά παράθυρα διαδρομής ([**e**$_i$ l_i])
2. Μέγιστος χρόνος κίνησης (***MVRT*** $_V$)
3. Μέγιστος χρόνος ολοκλήρωσης μιας (***DRT***$_i$)
4. Μέγιστη ικανότητα μεταφοράς οχήματος ($\boldsymbol{Q}_v$)

Το μικτό γραμμικό και ακέραιο πρόγραμμα είναι το εξής:

$$\min \sum_{v \in V} \sum_{i \in P^{all}} \sum_{j \in P^{all}} x_{ij}^{v} c_{ij} \qquad (1)$$

ST:

$$\sum_{v \in V} \sum_{j \in P^{all}} x_{ij}^{v} = 1, i \in P^{+} \qquad (2)$$

$$\sum_{j \in P^{all}} x_{ij}^{v} - \sum_{j \in P^{all}} x_{ji}^{v} = 0, i \in P, v \in V \qquad (3)$$

$$\sum_{j \in P^{+}} x_{ij}^{v} = 1, i \in VSD, v \in V \qquad (4)$$

$$\sum_{i \in P^{-}} x_{ij}^{v} = 1, j \in VED, v \in V \qquad (5)$$

$$\sum_{j \in P^{all}} x_{ij}^{v} - \sum_{j \in P^{all}} x_{j(N+i)}^{v} = 0, i \in P^{+}, v \in V \qquad (6)$$

$$T_i^{v} + s_i + \tau_{i(i+N)} \leq T_{(i+N)}^{v}, i \in P^{+}, v \in V \qquad (7)$$

$$x_{ij}^{v} = 1 \Rightarrow T_i^{v} + s_i + \tau_{ij} <= T_j^{v}, i, j \in P \ , v \in V \qquad (8)$$

$$x_{ij}^{v} = 1 \Rightarrow T_i^{v} + s_i + \tau_{ij} <= T_j^{v}, i \in VSD, j \in P^{+} \qquad (9)$$

$$x_{ij}^{v} = 1 \Rightarrow T_i^{v} + s_i + \tau_{ij} <= T_j^{v}, j \in VED, i \in P^{-} \qquad (10)$$

$$x_{ij}^{v} = 1 \Rightarrow L_i^{v} + d_j = L_j^{v}, i \in P \ , j \in P \ , v \in V \qquad (11)$$

$$x_{ij}^{v} = 1 \Rightarrow L_i^{v} + d_j = L_j^{v}, i \in VSD, j \in P^{+}, v \in V \qquad (12)$$

$$L_i^{v} = 0, i \in VSED \qquad (13)$$

$$e_i \le T_i^v \le l_i, i \in P, v \in V \tag{14}$$

$$e_i \le T_i^v \le l_i, i \in VSD \;, v \in V \tag{15}$$

$$e_i \le T_i^v \le l_i, i \in VED \;\;, v \in V \tag{16}$$

$$d_i \le L_i^v \le Q^v, i \in P^+, v \in V \tag{17}$$

$$T_{(i+N)}^v - T_i^v \le DRT_i, i \in P^+, v \in V \tag{18}$$

$$T_i^v \le MVRT_v, i \in P^{all}, v \in V \tag{19}$$

$$x_{ij}^v, binary$$

Η αντικειμενική συνάρτηση (1) ελαχιστοποιεί το συνολικό κόστος το οποίο παράγεται από όλους τους μοναχούς.

Ο περιορισμός (2) εγγυάται ότι ένας μοναχός παραλαμβάνει ένα αντικείμενο από μία συγκεκριμένη θέση και το παραδίδει σε μία άλλη συγκεκριμένη θέση.

Ο περιορισμός (3) εγγυάται ότι ο αριθμός των μοναχών που καταλήγουν σε μία θέση i είναι ίσος με τον αριθμό των μοναχών που φεύγουν από την θέση i.

Ο περιορισμός (4) εγγυάται ότι μετά το αρχικό σημείο αναφοράς για τους μοναχούς ο κάθε ένας από αυτούς θα πάει για παραλαβή αντικειμένου σε ένα συγκεκριμένο (μοναδικό) σημείο.

Ο περιορισμός (5) εγγυάται ότι είναι μία συγκεκριμένη θέση στην οποία ο κάθε μοναχός θα παραδώσει το αντικείμενο πριν καταλήξει στο τελικό σημείο αναφοράς. Ο περιορισμός (6) είναι συνδυασμός περιορισμών και εννοεί ότι ο ίδιος μοναχός επισκέπτεται τον κόμβο i ,i+n.

Οι περιορισμοί (7-10) είναι περιορισμοί προτεραιότητας και μας εξασφαλίζουν ότι επισκεφτήκαμε τον κόμβο i πριν τον κόμβο i+n καθώς επίσης και την συμβατότητα μεταξύ διαδρομών και χρονοδιαγράμματος.

Ο περιορισμός (10-13) εξασφαλίζει τη συνέπεια των μεταβλητών φορτίων για κάθε μεταφορά.

Ο περιορισμός (14) εξασφαλίζει ότι κάθε μεταφορά δεν παραβιάζει τα χρονικά περιθώρια.

Οι περιορισμοί (15-16) εγγυώνται ότι δεν παραβιάζονται τα χρονικά περιθώρια που έχει κάθε μοναχός.

Ο περιορισμός (17) εξασφαλίζει ότι η μέγιστη ικανότητα μεταφοράς που έχει κάθε μοναχός δεν παραβιάζεται.

Ο περιορισμός (18) διατηρεί τον μέγιστο χρόνο ολοκλήρωσης μιας διαδρομής.

Ο περιορισμός (19) διατηρεί τον μέγιστο χρόνο διαδρομής κάθε μοναχού.

3.3 ΕΠΙΛΟΓΗ ΑΛΓΟΡΙΘΜΟΥ

Ο απλός ευρετικός αλγόριθμος εισαγωγής

Ο απλός αλγόριθμος εισαγωγής που έχει χρησιμοποιηθεί είναι μια παραλλαγή αυτού που χρησιμοποιήθηκε από τους Jaw(6). Ο βασικός στόχος εκείνου του αλγόριθμου είναι να δημιουργήσει (συγκεκριμένες διαδρομές που ορίζονται σε συγκεκριμένα οχήματα). Μια σύντομη περιγραφή του "απλού ευρετικού" αλγορίθμου εισαγωγής ακολουθεί στην επόμενη παράγραφο:

Στην αρχή, ο αλγόριθμος δημιουργεί τις πιθανές διαδρομές για κάθε μοναχό. Μετά από αυτό ο αλγόριθμος βρίσκει για κάθε διαδρομή την καλύτερη θέση από άποψη κόστους που έχουν οι μοναχοί σε σχέση με την διαδρομή. Εάν βρεθεί αυτή η θέση τότε η ανάγκη αυτής της διαδρομής παρεμβάλλεται σε εκείνη τη θέση. Εάν δεν καταστεί δυνατό να βρεθεί τουλάχιστο ένας μοναχός στη διαδρομή του οποίου θα μπορεί να παρεμβληθεί η συγκεκριμένη διαδρομή τότε η διαδρομή αυτή απορρίπτεται. Στο τέλος της εκτέλεσης του αλγορίθμου έχουν οριστεί συγκεκριμένες διαδρομές σε συγκεκριμένα οχήματα. Ένας ψευδοκώδικας που περιγράφει τον "απλό ευρετικό" αλγόριθμο εισαγωγής ακολουθεί παρακάτω:

Step 0: For each vehicle v (v=1,2,3.....|V|)

Build an empty route for each vehicle

Step 1: For each trip request j (j=1,2,3....n)

For each vehicle v (v=1,2,3.....|V|)

Find the best position to insert in the vehicle route that produces the minimum cost for that vehicle route.

If that position has been found then name that cost $Cost_v$

From all costs $Cost_i$ ***(i=1,2,3...,|v|)that have been found select the minimum one***

Assign that trip request to the specific vehicle that the produced cost was the minimum, otherwise reject that trip request

3.5 ΣΥΜΠΕΡΑΣΜΑΤΑ

Καταλήγοντας σε αυτή την ενότητα είμαστε έτοιμοι να τρέξουμε τα παραδείγματα για να πάρουμε αποτελέσματα. Έχουμε κάνει την επιλογή των επιμέρους προγραμμάτων που θα χρησιμοποιηθούν (Matlab, Excel). Αυτά είναι τα προγράμματα που θεωρούμε είναι πιο λειτουργικά και πιο εύχρηστα, και αναμένουμε να πάρουμε τα ανάλογα αποτελέσματα.

Όσο αφορά τον απλό ευρετικό αλγόριθμο το είχαμε σχεδιάσει εξαρχής ότι θα τον επιλέξουμε αφού πάνω σε αυτόν βασίζεται το όλο σκεπτικό της εργασίας αυτής. Εμείς απλός θα πρέπει να φέρουμε το πρόβλημά μας στη μορφή που ο αλγόριθμος δίνει αποτελέσματα και να τα συγκρίνουμε με τα αποτελέσματα ενός εξακριβωμένα ορθού κώδικα της AMPL.

ΚΕΦΑΛΑΙΟ 4
ΔΕΔΟΜΕΝΑ-ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

ΚΕΦΑΛΑΙΟ 4: ΔΕΔΟΜΕΝΑ-ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

Από την αρχή, ξεκινήσαμε αυτή την εργασία έχοντας στο μυαλό μας, τη δομή του προβλήματος και υπολογίζοντας περίπου το μέγεθός του γι' αυτό άλλωστε και επιλέξαμε τον συγκεκριμένο ευρετικό αλγόριθμο, προκειμένου να οδηγηθούμε στα ορθά αποτελέσματα.

Τελικά σήμερα που είμαστε έτοιμοι να τρέξουμε το πρόβλημα δεν υπάρχουν πραγματικά στοιχεία, ως εκ τούτου είμαστε αναγκασμένοι να το εφαρμόσουμε σε τυχαίες τιμές. Για να δούμε αν η όλη διαδικασία που θα εφαρμόσουμε έχει τα θεμιτά αποτελέσματα, το πρόβλημα θα το τρέξουμε τυχαία για 4 ,5, και για 10 αντικείμενα.Τις τυχαίες θέσεις των αντικειμένων θα τις πάρουμε τρέχοντας τον κώδικα με την ονομασία Myrandint της ***Matlab***. Για το παράδειγμα των τεσσάρων αντικειμένων θα χρειαστεί να γίνει εφαρμογή του και σε ένα κώδικα της ***AMPL*** ο οποίος είναι εξακριβωμένα ορθός, ώστε να μπορούμε να συγκρίνουμε τα αποτελέσματά μας. Αν το κόστος που θα βρούμε από την ***AMPL,*** συμπίπτει με το κόστος που βγάζουμε από τον ευρετικό αλγόριθμο που χρησιμοποιούμε, τότε καταλήγουμε ότι η σκέψη χρησιμοποίησης αυτής της μεθόδου ήταν επιτυχής.

4.1 Ένας μοναχός - τέσσερα αντικείμενα

4.1.1 Απλός ευρετικός αλγόριθμος

Μεταφερόμενοι σε περιβάλλον ***Matlab*** θέλουμε να δημιουργήσουμε ένα τυχαίο πίνακα, έστω q=(5X3) με τιμές από 0-100, όπου η πρώτη γραμμή θα έχει μηδενικές τιμές αφού όπως έχουμε προαναφέρει ο κόμβος ένα θα είναι το αρχικό σημείο αναφοράς από το οποίο θα φεύγουν οι μοναχοί και στο οποίο θα καταλήγουν. Έτσι ο κόμβος ένα θεωρούμε ότι έχει κόστος (αξία) μηδενικό καθώς επίσης το σημείο αναφοράς έχει συντεταγμένες Χ=0 ,Ψ=0. Αυτό το καταφέρνουμε με την εντολή ***q=myrandin (4,3,0:100)*** και ακολούθως αφού δημιουργήσουμε τον τυχαίο πίνακα ***q*** παρεμβαίνουμε σε αυτόν για να μηδενίσουμε την πρώτη γραμμή.

ΠΙΝΑΚΑΣ 4: Τυχαίος πίνακας q 1^{ου} παραδείγματος

Κόμβος - Κειμήλιο	Κόστος	Άξονας-Χ	Άξονας-Ψ
1(Start depot - End depot)	0	0	0
2	23	46	79
3	61	43	93
4	49	82	74
5	90	44	17

Ακολούθως αφού δημιουργήσουμε τον πίνακα ***q*** χρησιμοποιούμε το πρόγραμμα ''***calc***'' στη ***Matlab*** που έχουμε φτιάξει για να βρούμε όλα τα τόξα που προκύπτουν από τους κόμβους του παραδείγματός μας. Θέλουμε να βρούμε τα κόστη από κόμβο σε κόμβο καθώς επίσης και τους χρόνους που ένας μοναχός χρειάζεται για να διανύσει την κάθε απόσταση. Ο καινούργιος πίνακας που θα δημιουργήσουμε έστω ονομάζεται ***R*** και λαμβάνεται εκτελώντας την παρακάτω εντολή στη ***Matlab***.

R=calc (q,0.1) όπου 0.1 είναι ο χρόνος σε λεπτά που υποθέτουμε ότι ένα όχημα –μοναχός χρειάζεται για να διανύσει απόσταση ίση με μια μονάδα απόστασης. Το 0.1 είναι για τη συγκεκριμένη περίπτωση ο χρόνος που απαιτείται από ένα μοναχό σε λεπτά για να διανύσει μία μονάδα απόστασης για παράδειγμα ένα μέτρο. Καθότι για κάθε μοναχό θα μπορούσαμε να υπολογίσουμε την ταχύτητα κίνησής του, άρα και τον χρόνο που χρειάζεται για να διανύσει ένα μέτρο αν η απόσταση που έχουμε είναι σε μέτρα.

Ο ευρετικός αλγόριθμος του ***Pick up and delivery*** που εμείς χρησιμοποιήσαμε σε αυτή τη διπλωματική δεν είναι σε θέση να διαβάσει τα δεδομένα μας από τον πίνακα ***R*** έτσι εμείς τώρα καλούμαστε να το φέρουμε σε αναγνωρίσιμη μορφή .Αυτό μπορεί να γίνει χρησιμοποιώντας τις μακροεντολές από το ***EXCEL*** και το παίρνουμε σε μορφή που θα μπορεί να χρησιμοποιηθεί από τον αλγόριθμό μας. Η μορφή αυτή παραθέτεται στο παράρτημα της διπλωματικής αναλυτικά.

Στην μορφή που έχουμε φέρει το πρόβλημά μας ο αλγόριθμος τώρα μπορεί να το διαβάσει. Στις απαιτήσεις (Demands) που έχουμε στο συγκεκριμένο πρόβλημα βάζουμε πρώτα αυτές με τη μεγαλύτερη αξία για εμάς, δηλαδή έχουμε βάλει πρώτα το DEM 5-1 όπου ως τόξο έχει το μεγαλύτερο κόστος από όλα τα αντικείμενα. Αυτό το κάνουμε διότι ο ευρετικός αλγόριθμος που χρησιμοποιήσαμε είναι σχεδιασμένος να μας δίνει το min 1/cost στο μικρότερο δυνατό χρόνο. Επομένως εμείς του βάζουμε την προτεραιότητά μας σύμφωνα με την αξία του κάθε κόμβου, και το πρόγραμμα σύμφωνα με τον χρόνο που ορίσαμε ότι έχει στη διάθεσή του ο κάθε μοναχός μας δίνει την βέλτιστη λύση. Στο συγκεκριμένο πρόβλημα έχουμε ένα μοναχό που μπορεί να τρέχει 45 λεπτά κατά τη διάρκεια της όλης διαδικασίας εκκένωσης και μέσα σε αυτό το χρονικό διάστημα θα πρέπει να παραλάβει και να παραδώσει εάν είναι δυνατό όλα τα αντικείμενα με προτεραιότητα όπως αναφέραμε σε αυτά με την μεγαλύτερη αξία.

Τρέχοντας τώρα και τον ευρετικό αλγόριθμο έχουμε πάρει τα εξής αποτελέσματα:

ΠΙΝΑΚΑΣ 5: Τελικά αποτελέσματα 1ου παραδείγματος

STATISTICS

Total Cost	***0.132***
Total Time	***43.44***
Total Demand SP Cost	***0.09139***
Total Demand real Cost	***0.09603***

4.1.2 AMPL(CPLEX)

Ακολούθως αφού φέρουμε τα δεδομένα στη μορφή που η AMPL μπορεί να αναγνωρίσει (Παράρτημα), τρέχουμε το πρόβλημα και εδώ και κάνουμε σύγκριση αποτελεσμάτων.

ΠΙΝΑΚΑΣ 6: Επαλήθευση αποτελεσμάτων 1ου παραδείγματος

CPLEX 9.1.0: timing=1

Times (seconds):
Input = 0.003
Solve = 0.177973
Output = 0.002
CPLEX 9.1.0: optimal (non-)integer solution; objective 0.1326
2929 MIP simplex iterations
386 branch-and-bound nodes
1 integer variables rounded (maxerr = 1.73195e-14).
Assigning integrality = 0 might help.
Currently integrality = 1e-05.
xvij :=

1 1 2 1
1 2 3 1
1 3 4 1
1 4 5 1
1 5 6 1
1 6 8 1
1 7 ved1 1
1 8 7 1
1 vsd1 1 1
;

Tvi :=

1 1 9.1417

1 2 27.0943

1 3 32.4325

1 4 40.283

1 5 46

1 6 47

1 7 49

1 8 48

1 ved1 50

;

Lvi :=

4.1.3 Συμπεράσματα και ανάλυση αποτελεσμάτων

Λαμβάνοντας τα αποτελέσματα και από τις δύο μεθόδους που χρησιμοποιήσαμε στο παράδειγμά μας, με ικανοποίηση βλέπουμε ότι τα συνολικά κόστη που πρωτίστως μας αφορούν είναι όμοια (0.132). Η πιστοποίηση που ζητούσαμε για τον αλγόριθμό μας έρχεται μέσα από την *AMPL* και τα αποτελέσματα που μας έδωσε. Τελικά αυτό που ζητούσαμε σε αυτή την διπλωματική το επιτύχαμε, καταφέρνοντας όπως το παράδειγμα δείχνει να διασώσουμε όλα τα αντικείμενα στον χρόνο που είχαμε καθώς επίσης μεγιστοποιήσαμε το κόστος.

Αφού τώρα έχουμε δοκιμάσει την μέθοδό μας και το αποτέλεσμα είναι εξακριβωμένα ορθό, είμαστε σε θέση να αναφέρουμε και την διαδρομή που ο μοναχός ενδείκνυται να ακολουθήσει. Η διαδρομή φαίνεται ξεκάθαρα στα ολοκληρωμένα αποτελέσματα που βρίσκονται στο παράρτημα. Αρχικά ξεκινά από τον κόμβο 1 (start point),πάει στον κόμβο 4 για να παραλάβει το αντικείμενο, μετά στον κόμβο 5 για να παραλάβει και το επόμενο αντικείμενο, καταλήγοντας τελικά και πάλι στο start point όπου έχουμε ορίσει ως σημείο παράδοσης των αντικειμένων. Ακολούθως μεταφέρεται στον κόμβο 2 και 3 αντίστοιχα για παραλαβή των κειμηλίων και επιστρέφει ξανά στον κόμβο 1 όπου ολοκληρώνεται η διαδρομή του με την παράδοση και των δύο αντικειμένων. Παρατηρούμε επίσης ότι ο μοναχός ολοκληρώνει την διαδρομή του στα προβλεπόμενα χρονικά περιθώρια, άρα και εδώ ο αλγόριθμος μας βλέπουμε ότι λειτουργεί χωρίς κανένα πρόβλημα.

4.2 Ένας μοναχός - πέντε αντικείμενα

Έχοντας υπόψη τα βήματα που ακολουθήσαμε στο προηγούμενο παράδειγμα φτιάχνουμε και εδώ τον αρχικό πίνακα q, μόνο που σε αυτό το παράδειγμα θα αρκεστούμε μόνο στα αποτελέσματα του απλού ευρετικού αλγόριθμου αφού πλέον ξέρουμε ότι τα αποτελέσματα που θα πάρουμε θα είναι ορθά:

ΠΙΝΑΚΑΣ 7: Τυχαίος πίνακας q 2ου παραδείγματος

Κόμβος - Κειμήλιο	Κόστος	Άξονας-X	Άξονας-ψ
1(Start depot - End depot)	0	0	0
2	35	34	74
3	61	82	17
4	56	44	40
5	90	62	94
6	95	79	92

ΠΙΝΑΚΑΣ 8: Τελικά αποτελέσματα 2ου παραδείγματος

STATISTICS

Total Cost	***0.109317***
Total Time	***0.055887***
Total Demand SP Cost	***0.107268***
Total Demand real Cost	***0.451***

4.2.1 Συμπεράσματα και ανάλυση αποτελεσμάτων

Στα ολοκληρωμένα αποτελέσματα στο παράρτημα βλέπουμε ότι ο μοναχός όντως έχει πάρει τα πολυτιμότερα αντικείμενα στον χρόνο που του είχαμε διαθέσει. Περιορίζοντάς του τον χρόνο κίνησης σε 45 λεπτά δεν έχει την δυνατότητα να πάρει όλα τα αντικείμενα, παρά μόνο τα πιο πολύτιμα. Στη συγκεκριμένη περίπτωση βλέπουμε ότι ο μοναχός μας κάνει δύο διαδρομές. Στη μία παίρνει το αντικείμενο από τον κόμβο 4 και ακολούθως μεταφέρεται στον κόμβο 3 για παραλαβή του επόμενου αντικειμένου και τελικά στον κόμβο 1 για παράδοση των 3 και 4 αντικειμένων.

Ακολούθως και έχοντας γνώση του χρόνου που του απομένει, επιλέγει να παραλάβει το αντικείμενο του κόμβου 6 και 5 αφήνοντας το αντικείμενο του κόμβου 2 το οποίο έχει και την μικρότερη αξία. Εδώ θα πρέπει να αναφέρουμε ότι ο μοναχός μας είχε ικανότητα μεταφοράς εώς δύο αντικείμενα ταυτόχρονα, ενώ τα αντικείμενα που είχαμε δεν απαιτούσαν χρόνους παραλαβής και παράδοσης. Επίσης αυτό που μας ενδιαφέρει είναι ο συνολικός χρόνος κίνησης του μοναχού (total time=43.86) ο οποίος είναι μέσα στα όρια που θέσαμε καθώς επίσης και το πραγματικό κόστος αντικειμένων που έχουμε παραλάβει(total cost=0.109317) που θα όπως υπολογίζαμε είναι η μικρότερη συνολική απόσταση που μπορούσαμε να διανύσουμε, δηλαδή το μέγιστο κόστος που μπορούσαμε να έχουμε. Κάτι άλλο το οποίο θα μπορούσε να μας φανεί χρήσιμο είναι η αξία των αντικειμένων που παραλάβαμε (total demands SP cost=0.055887) όπου φαίνεται εδώ η ακριβής αξία αντικειμένων που διασώσαμε.

4.3 Τρεις μοναχοί - δέκα αντικείμενα

Εφαρμόζοντας τα βήματα από το προηγούμενο παράδειγμα, φτιάχνουμε τον αρχικό μας πίνακα q:

ΠΙΝΑΚΑΣ 9: Τυχαίος πίνακας q 3ου παραδείγματος

Κόμβος - Κειμήλιο	Κόστος:	Άξονας-Χ	Άξονας-Ψ
1(Start depot - End depot)	0	0	0
2	23	93	64
3	46	74	14
4	49	17	20
5	54	40	20
6	76	94	60
7	61	92	27
8	58	41	20
9	82	90	31
10	90	5	75
11	78	35	44

Επόμενο βήμα είναι η εύρεση όλων των τόξων και των 11 κόμβων με την χρήση του προγράμματος 'calc' στη ***MATLAB*** .Όπου t=0.1 και σε αυτή την περίπτωση.

Τελικά αφού φέρουμε τον πίνακα R στη μορφή που πρέπει για να μπορεί να μπει στον αλγόριθμο που χρησιμοποιούμε τρέχουμε το πρόγραμμα για να πάρουμε τις λύσεις: Στην συγκεκριμένη περίπτωση θα πρέπει να αναφέρουμε ότι θα χρησιμοποιήσουμε 3 οχήματα (μοναχούς), όπου ο καθένας θα έχει ικανότητα μεταφοράς 2 αντικειμένων και χρόνο κίνησης 30 λεπτά. Τα κειμήλια μας επίσης δεν θα έχουν επιπρόσθετους χρόνους παραλαβής –

παράδοσης. Θα ζητούμε τελικά όσο περισσότερα αντικείμενα μπορούν να μεταφέρουν στους προβλεπόμενους χρόνους με κριτήριο την μέγιστη αξία τους.

Τρέχοντας τελικά το πρόγραμμα παίρνουμε τα παρακάτω αποτελέσματα:

ΠΙΝΑΚΑΣ 10: Τελικά αποτελέσματα 3ου παραδείγματος

STATISTICS

Total Cost	***0.2113***
Total Time	***78.18***
Total Demand SP Cost	***0.121846***
Total Demand real Cost	***0.207142***

4.3.1 Συμπεράσματα και ανάλυση αποτελεσμάτων

Διαβάζοντας τα αποτελέσματα που παίρνουμε πλέον είμαστε σίγουροι ότι ο αλγόριθμός μας δουλεύει και σε αυτή την περίπτωση αφού στον καθορισμένο χρόνο που του δώσαμε κατάφερε να μας βρει το min 1/cost δηλαδή το μέγιστο κόστος αντικειμένων που μπορεί να διασώσει στα χρονικά παράθυρα που του θέσαμε. Μπορούμε να διακρίνουμε και την απόσταση που διάνυσε το κάθε όχημα (μοναχός).

Ο μοναχός 1 ξεκίνησε από τον κόμβο 1, και μετέβη στον κόμβο 9 από όπου παρέλαβε το αντικείμενο, και στη συνέχεια πήγε για παραλαβή του αντικειμένου 10. Αφού η ικανότητα μεταφοράς που είχε δεν του επέτρεπε για να πάρει άλλο αντικείμενο, κατευθύνθηκε στον κόμβο 1 όπου παρέδωσε τα αντικείμενα που είχε διασώσει. Ο χρόνος κίνησης που του απέμενε δεν ήταν αρκετός για ακόμα μια διαδρομή έτσι αρκέστηκε στην μια και μοναδική όπου διέσωσε τα κειμήλια των κόμβων 9 και 10.

Ο μοναχός 2 τώρα ξεκινάει και κατευθύνεται στον κόμβο 6 και αμέσως μετά στον 11 από όπου παραλαμβάνει τα κειμήλια. Εν συνεχεία κατευθύνεται στο Start depot στον κόμβο 1 δηλαδή για την παράδοση των αντικειμένων. Αφού γίνει η παράδοση βλέπει ότι ο χρόνος κίνησης που του απομένει δεν είναι αρκετός για άλλη μία διαδρομή έτσι ολοκληρώνει και αυτός την προσπάθειά του για διάσωση όσο το δυνατό περισσότερων αντικειμένων.

Η προσπάθεια του μοναχού 3 αρχίζει με τα αντικείμενα των κόμβων 4 και 5 τα οποία σε σύντομο χρονικό διάστημα, αφού οι αποστάσεις μεταξύ τους είναι μικρές παραλαμβάνει και παραδίδει στον κόμβο 1. Ακολούθως μεταβαίνει στον κόμβο 8 και 7 από όπου παραλαμβάνει τα αντίστοιχα κειμήλια τα οποία μεταφέρει και πάλι στον κόμβο 1.

Τα αντικείμενα 2 και 3 δεν κατέστη δυνατό να τα περισώσουμε αφού ο χρόνος που είχαμε στη διάθεσή μας δεν ήταν αρκετός. Εξάλλου η αξία των αντικειμένων αυτών ήταν μικρότερη από τα διασωθέντα γι'αυτό και είμαστε ικανοποιημένοι από τα αποτελέσματα του προγράμματος αφού καταφέραμε να προστατεύσουμε τα πολυτιμότερα αντικείμενα.

Γενικά τώρα και πάλι βλέπουμε ότι ο συνολικός χρόνος κίνησης που είχαν τα οχήματά μας είναι 78.18 min που είναι μέσα στους περιορισμούς που θέσαμε καθώς ο κάθε μοναχός δεν ξεπέρασε τα 30 min χρόνο κίνησης που είχε διαθέσιμο. Το συνολικό κόστος διαδρομής είναι 0.21 και είναι και πάλι το μικρότερο δυνατό όπως επίσης και η αξία των αντικειμένων που διασώθηκαν είναι η μέγιστη.

ΚΕΦΑΛΑΙΟ 5
ΤΕΛΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ

ΚΕΦΑΛΑΙΟ 5: ΤΕΛΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ

Στις ενότητες της παρούσας διπλωματικής εργασίας που έχουν προηγηθεί παρουσιάσαμε ένα ευρετικό αλγόριθμο εύρεσης της βέλτιστης διαδρομής που θα πρέπει ένα όχημα ή ένας στόλος οχημάτων να ακολουθήσουν, ώστε να έχουμε την μεγαλύτερη συνολική αξία αντικειμένων μετά την συλλογή αυτών. Κατά την προσπάθεια πλήρους παρουσίασης του αλγορίθμου αυτού (ευρετικός ***Pick up and Delivery***) αλλά και επιβεβαίωσης της ορθής λειτουργίας του, πραγματοποιήσαμε μία σειρά πειραμάτων, τρία εκ των οποίων έχουμε προαναφέρει σε προηγούμενη ενότητα. Ο απλός ευρετικός αλγόριθμος που χρησιμοποιήσαμε έχει κατασκευαστεί για την εύρεση της min διαδρομής που θα πρέπει να διανύσει ένα ή περισσότερα οχήματα έτσι ώστε να ικανοποιήσουν μία σειρά αιτημάτων. Εμείς αξιοποιήσαμε τη χρησιμότητα του αλγορίθμου αυτού αντιλαμβανόμενοι το δικό μας πρόβλημα σαν παρόμοιου τύπου με αυτά που ο συγκεκριμένος αλγόριθμος επιλύει.

Στην αρχή έπρεπε να κατανοήσουμε το πρόβλημά μας το οποίο ζητούσε την μέγιστη εξασφάλιση και διάσωση των κειμηλίων σε περίπτωση εκτάκτου ανάγκης με βάση την αξία τους. Αμέσως αντιλαμβανόμαστε ότι πρόκειται για πρόβλημα παραλαβής – παράδοσης. Εμείς έχοντας εις γνώση μας τον απλό ευρετικό αλγόριθμο φέραμε το πρόβλημά μας στη μορφή που έπρεπε ώστε να μπορεί να λειτουργήσει ο αλγόριθμος. Θέλαμε το max της αξίας των αντικειμένων άρα αφού ο αλγόριθμος δούλευε με το min της απόστασης έπρεπε να πάρουμε το αντίστροφο του κόστους δηλαδή το κάναμε 1/cost για να ζητούμε το min. Έχοντας υπόψη μας και τους χρόνους που χρειάζεται για να διανύσουμε την κάθε απόσταση, μπορούσαμε έτσι να δοκιμάσουμε την λειτουργικότητα του αλγορίθμου και σε αυτό το πρόβλημα. Σαν γενική εικόνα μπορούμε να μεταφέρουμε την ικανοποίησή μας από τα αποτελέσματα των πειραμάτων, αφού ήταν όπως τα περιμέναμε. Η πιστοποίηση βέβαια της λειτουργικότητας του αλγορίθμου ήρθε μέσα από την *AMPL* που απλά μας επιβεβαίωσε τα δικά μας αποτελέσματα.

Από τα αποτελέσματα που παρατίθενται παραπάνω, παρατηρούμε ότι ο αλγόριθμος που χρησιμοποιούμε είναι σε θέση να μας δώσει την βέλτιστη διαδρομή. Ελέγχει ορθά την ικανότητα μεταφοράς των οχημάτων (μοναχών) καθώς επίσης ανταποκρίνεται και στους χρονικούς περιορισμούς που έχουν στην κίνησή τους. Παράλληλα καταφέρνει να εστιάζει στη διάσωση των πολυτιμότερων αντικειμένων αφού δίνοντας τους εμείς προτεραιότητα στη μορφή που το πρόγραμμα διαβάζει, καταφέρνει πάντα να διασώζει αυτά με το μεγαλύτερο κόστος. Στα παραδείγματα που έχουμε παρουσιάσει δεν έχουμε βάλει επιπρόσθετους χρονικούς περιορισμούς στα απαιτούμενα αντικείμενα που θέλουμε να διασώσουμε. Μέσα στα πειράματα που τρέξαμέ όμως σε μερικές απαιτήσεις κειμηλίων που είχαμε, βάλαμε και χρονικά περιθώρια παραλαβής και παράδοσης αντικειμένων και ο αλγόριθμός μας ανταποκρίθηκε και πάλι με επιτυχία.

Τελικά το πρόβλημα που επιλύσαμε μέσω αυτού του αλγορίθμου δεν ήταν πραγματικό, αφού στοιχεία δεν μπορέσαμε να έχουμε. Η λειτουργικότητα όμως του αλγόρίθμου μας και σε αυτού του είδους προβλήματα μας δίνει ενθαρρυντικά μηνύματα για την χρησιμότητά του και σε πραγματικά προβλήματα.

Το πρόβλημα που εμείς επιλύσαμε αναφερόταν σε δύο χώρους, τον –Χ και –Ψ. Σε μεταγενέστερο στάδιο θα ήταν θεμιτό να ασχοληθούμε με το ίδιο πρόβλημα σε τρεις διαστάσεις. Θα ήταν σίγουρα πιο ρεαλιστικό από άποψης χώρου. Με το δικό μας αλγόριθμο μπορούν να επιλυθούν τέτοιου είδους προβλήματα, αλλά με μικρότερη ακρίβεια αφού για κάποιο κειμήλιο που βρίσκεται σε ένα ύψος -Ζ μπορούμε να προσθέσουμε όπως προαναφέραμε χρονικά παράθυρα παραλαβής .

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] H.N. Psaraftis.A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem.Tranportation Science 14(1980) 130-154

[2] Psaraftis H "An exact algorithm for the single vehicle many to- many dial-a-ride problem with time windows". Transportation Science 17 ,1983,(3): 351-357.

[3] DumasY, Desrosiers J and Soumis F "The pickup and delivery problem with time windows". European J ournal of Operational Research ,1991,54: 7—22.

[4] Desrosiers J, Dumas Y and Soumis F "A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows". American J ournal of Mathematical and Management Sciences ,1986,6: 301-325.

[5] Jaw, J., Odoni, A. R., Psaraftis, H. N., & Wilson, N. H. M. (1986). A heuristic algorithm for the multi-vehicle advance-request dial-a-ride problem with time windows. *Transportation Research B*, *20*, 243–257.

[6] Sexton, Bodin. Optimizing Single Vehicle many to many operations with Desired Delivery Times I Scheduling. *Transportation Science Vol 19. No 4*, November 1985.

[7] Sexton,Bodin .Optimizing Single Vehicle many to many operations with Desired Delivery Times II Routing. *Transportation Science Vol 19. No 4*, November 1985..

[8] Paolo Toth and Daniel Vigo. Heuristic Algorithms for the handicapped persons Transportation Problem. *Transportation Science Vol 31,1997, No 1.*

[9] Bodin, L. D., Golden, B. L., Assad, A. e Ball, M. (1983). “Routing and Scheduling of Vehicles and Crews”. *Computeres and Operations Research,* 10, 69-211.

[10] Ioachim, I., Desrosiers, J., Dumas, Y., Solomon, M.M., Villeneuve, D., 1995. A request clustering algorithm for door-to-door handicapped transportation. Transportation Science 29, 63–78.

[11] Savelsbergh, M. W. P. (1992). The vehicle routing problem with time windows: minimizing route duration. *ORSA Journal on Computing*, *4*, 146–154.

[12] Borndorfer, R., Klostermeier, F., Grφtschel, M., & Kuttner, C. (1997). Telebus Berlin: vehicle scheduling in a dial-a-ride system, Technical report SC 97-23, Konrad-Zuse-Zentrum fόr Informationstechnik, Berlin.

[13] Healy P and Moll R "A new extension of local search applied to the Dial-A-Ride Problem". European J ournal of Operational Research ,1995,83: 83-104.

[14] Thompson PM and Psaraftis HN (1993)’’Cyclic transfer algorithms for multivehicle routing and sheduling problems’’. Operations Research 41 (5) :935-946.

[15] Desrosiers J, Dumas Y and Soumis F. Taillefer S and Villeneuve D (1991)’’An algorythm for mini-clustering in handicapped transport’’.Report G-91-01 .Ecole des Hautes Etudes Commerciales, Mondreal.

[16] Van Der Bruggen LJJ, Lenstra JK and Schuur PC(1993)''Variable-depth search for the single-vehicle pickup and delivery problem with time windows''. Transportation Science, 27 (3):298-311.

[17] Cordeau JF and Laporte G (2003)''A Tabu search heuristic for the static multi-vehicle dial-a-ride problem''. Transportation Research P art B 37:579-594.

[18] Toth P and Vigo D (1997)''Heuristic algorithms for the handicapped persons transportation problem''. Transportation Science 31(1):60-71.

[19] Bodin LD and Sexton T (1986)''The multi-vehicle subscriber dial-a-ride problem''. T IMS Studies in Management Science 2: 73-86.

ΠΑΡΑΡΤΗΜΑ [Α]

Κώδικας MATLAB (calc)

function [R,T]=calc(A,time);

t=(length(A)-1)*length(A); (Για να υπολογίσει το πλήθος των συνδυασμών όλων των κόμβων του πίνακα (A))

```
T=zeros(t,3);
x=1;
for j=1:length(A),
   for k=1:length(A),
      if k~=j
         T(x,1)=abs(A(j,2)-A(k,2));
         T(x,2)=abs(A(j,3)-A(k,3
         x=x+1;
      end
   end
end

for i=1:length(T),
   T(i,3)=sqrt(T(i,1)^2+T(i,2)^2);
end
R=zeros(t,5);
m=max(A(:,1));
c=1000;
for i=1:length(A),
   if A(i,1)>0
      if c>A(i,1)
         c=A(i,1);
      else
         c=c;
      end
   end
end
```

```
x=1;
for i=1:length(A),
  for j=1:length(A),
    if i~=j
      R(x,1)=i;
      R(x,2)=j;
      R(x,3)=time*T(x,3);
      if A(i,1)>c
        add=A(i,1)-c+1;
        R(x,4)=1/(A(j,1)-A(i,1)+add);
        R(x,5)=A(j,1)-A(i,1)+add;
      elseif A(i,1)==c.
        if A(j,1)-A(i,1)==0
          R(x,4)=1/(A(j,1)-A(i,1)+1);
          R(x,5)=A(j,1)-A(i,1)+1;
        else
          R(x,4)=1/(A(j,1)-A(i,1));
          R(x,5)=A(j,1)-A(i,1);
        end
      else
        R(x,4)=1/(A(j,1)-A(i,1));
        R(x,5)=A(j,1)-A(i,1);
      end
      if (A(i,1)==0 || A(j,1)==0)
        if A(i,1)==0
          R(x,4)=1/A(j,1);
          R(x,5)=A(j,1);
        else
          R(x,4)=1/A(i,1);
          R(x,5)=A(i,1);
        end
      end
      x=x+1;
    else
```

```
      x=x;
    end
  end
end
```

ΠΑΡΑΡΤΗΜΑ [B]

Κώδικας μακροεντολών EXSEL

```
Sub agatho_output()
'
' agatho_output Macro
'
'
Sheet1.Select
   n_lines = WorksheetFunction.CountA(Range("A3:A65536"))
   n_nodes = WorksheetFunction.CountIf(Range("A3:A65536"), "=1") + 1


 Sheet1.Cells(1, 8).Value = "Well done. File C:\agathoklis.txt created!"

Open "C:\agathoklis.txt" For Output As #1

   For i = 1 To n_nodes
      Print #1, "{"
      Print #1, "" & i
      Print #1, "Node" & i
      Print #1, "0"
      Print #1, "0"
      Print #1, "}"
   Next i
      Print #1, "ARCS"

   For i = 1 To n_lines
      Print #1, "{"
      Print #1, "Arc" & Cells(i + 2, 1).Value & "-" & Cells(i + 2, 2).Value
      Print #1, "" & Cells(i + 2, 1).Value
      Print #1, "" & Cells(i + 2, 2).Value
      Print #1, "" & Cells(i + 2, 4).Value
```

```
    Print #1, "" & Cells(i + 2, 3).Value
    Print #1, "}"
  Next i


Close #1
```

ΠΑΡΑΡΤΗΜΑ [Γ]

Κώδικας AMPL (CPLEX)

```
set VEHICLES; # vehicles
param vehicles > 0; # vehicles number
set VSD;      # start depot
param vsd > 0; # start depots number
set VED;      # end depot
param ved > 0; # end depots number
set VSED := VSD union VED; # start and end depots
param Q{VEHICLES} >0; # vehicle capacity always greater than 0
param MVRT{VEHICLES} >0; # vehicle capacity always greater than 0
param T > 0; # time parameter defines every time the running time of the system

param NtPickup > 0;
param Nt = NtPickup*2; # nodes number for time t.
param NtVSD = Nt + vsd;
param NtVED = Nt + ved;
param NtVSED = Nt + vsd + ved;

set Pplus;
set Pminus;
set P := Pplus union Pminus;
set PVSD := VSD union P;
set PVED := P union VED;
set PVSDplus := VSD union Pplus;
set PVEDplus := Pminus union VED;
set PVSED := PVSDplus union PVEDplus;
set PVEDPplus := VED union Pplus;
set PVSDPminus := VSD union Pminus;

param earlier{PVSED};
param later{j in PVSED}> earlier[j];
```

```
param drt{P};
param travel_time{PVSED,PVSED};
param  travel_cost{PVSED,PVSED};
param service_time{PVSED};
param load{P};


var xvij{v in VEHICLES,i in PVSED, j in PVSED} binary;
var Tvi{v in VEHICLES, i in PVSED} >= earlier[i], <= T;
var Lvi{v in VEHICLES, i in PVSED}>=0, <= (Nt div 2); # must be changed later

minimize total_cost_darpt_dumas : sum {v in VEHICLES,i in PVSED,j in PVSED}
travel_cost[i,j]*xvij[v,i,j];
# dumas constrains version
# ensure network flow constrains


subject to network_flow_6 {i in Pplus} : sum {v in VEHICLES,j in PVSED} xvij[v,i,j]
= 1;
subject to network_flow_mine {i in PVSED,v in VEHICLES} : xvij[v,i,i] = 0;
subject to network_flow_7 {v in VEHICLES,i in P} :  sum {j in PVSED}  xvij[v,i,j] -
sum {j in PVSED} xvij[v,j,i] = 0;
subject to network_flow_8 {v in VEHICLES, i in VSD} : sum {j in Pplus} xvij[v,i,j]
=1;
subject to network_flow_9 {v in VEHICLES, j in VED}  : sum {i in Pminus} xvij[v,i,j]
=1;
subject to network_flow_10 {v in VEHICLES,i in Pplus} : sum {j in PVSED} xvij[v,i,j]
- sum {j in PVSED} xvij[v,j,i+NtPickup] = 0;


#ensure service time constrains
subject to time_service1  {v in VEHICLES, i in Pplus} :
Tvi[v,i]+service_time[i]+travel_time[i,NtPickup+i] <= Tvi[v,NtPickup+i];
subject to time_service2  {v in VEHICLES, j in P, i in P} : Tvi[v,j]-Tvi[v,i]-
```

```
service_time[i]-travel_time[i,j] >= (xvij[v,i,j]-1)*9999999999;
subject to time_service3 {v in VEHICLES, i in VSD, j in Pplus} : Tvi[v,j]-Tvi[v,i]-travel_time[i,j] >= (xvij[v,i,j]-1)*9999999;
subject to time_service4 {v in VEHICLES, i in Pminus, j in VED} : Tvi[v,j]-Tvi[v,i]-service_time[i]-travel_time[i,j] >= (xvij[v,i,j]-1)*99999999;

# ensure time windows constrains
subject to time_windows1 {i in P, v in VEHICLES} : earlier[i]<= Tvi[v,i] <= later[i];
subject to time_windows2 {i in VSD, v in VEHICLES} : earlier[i]<= Tvi[v,i] <= later[i];
subject to time_windows3 {i in VED, v in VEHICLES} : earlier[i]<= Tvi[v,i] <= later[i];


#ensure load constrains
subject to load_balance1 {v in VEHICLES, j in Pplus,i in P} : Lvi[v,j]-Lvi[v,i]-load[j]>=(xvij[v,i,j]-1)*999999999;
subject to load_balance2 {v in VEHICLES, j in Pminus, i in P} : Lvi[v,j]-Lvi[v,i]+load[j-NtPickup]>=(xvij[v,i,j]-1)*99999999;
subject to load_balance3 {v in VEHICLES, i in VSD,j in Pplus} : Lvi[v,j]-Lvi[v,i]-load[j]>=(xvij[v,i,j]-1)*999999999;
```

ΠΑΡΑΡΤΗΜΑ [Δ]

Ένας μοναχός – τέσσερα αντικείμενα

ΠΙΝΑΚΑΣ 11 : Αποτελέσματα **Matlab (calc)**

Από κόμβο:	Προς κόμβο:	Χρόνος:	1/Κόστος:	Κόστος:
1	2	9.14	0.043478	23
1	3	10.25	0.016393	61
1	4	11.05	0.020408	49
1	5	4.72	0.011111	90
2	1	9.14	0.043478	23
2	3	1.43	0.026316	38
2	4	3.63	0.038462	26
2	5	6.20	0.014925	67
3	1	10.25	0.016393	61
3	2	1.43	1	1
3	4	4.34	0.037037	27
3	5	7.60	0.014706	68
4	1	11.05	0.020408	49
4	2	3.63	1	1
4	3	4.34	0.025641	39
4	5	6.85	0.014706	68
5	1	4.72	0.011111	90
5	2	6.20	1	1

5	3	7.60	0.025641	39
5	4	6.85	0.037037	27

ΠΙΝΑΚΑΣ 12: Ένα όχημα-Τέσσερα αντικείμενα-Input αλγορίθμου.

PREFERENCES
1 (αριθμός οχημάτων)
5 (αριθμός κόμβων)
20 (αριθμός τόξων)
4 (αριθμός αιτημάτων)
119
VEHICLES
{
MONK1
2 (ικανότητα μεταφοράς οχήματος)
1 (αρχικό σημείο αναφοράς)
0
1440
1 (τελικό σημείο αναφοράς οχήματος)
0
1440
50 (περιορισμός χρόνου κίνησης)
}
NODES
{
1
Node1
0
0
}
{
2
Node2
0
0
}
{
3
Node3
0
0
}
{
4
Node4
0
0
}
{
5
Node5
0
0
}
ARCS
{
Arc1-2
1 (από κόμβο)
2 (προς κόμβο)
0.043478 (1/κόστος)
9.14 (χρόνος)
}
{

Arc1-3
1
3
0.016393
10.25
}
{
Arc1-4
1
4
0.020408
11.05
}
{
Arc1-5
1
5
0.011111
4.72
}
{
Arc2-1
2
1
0.043478
9.14
}
{
Arc2-3
2
3
0.026316
1.43
}
{
Arc2-4
2
4
0.038462
3.63
}
{
Arc2-5
2
5
0.014925
6.20
}
{
Arc3-1
3
1
0.016393
10.25
}
{
Arc3-2
3
2
1
1.43
}
{
Arc3-4
3
4
0.037037
4.34
}
{
Arc3-5
3
5
0.014706
7.60
}
{
Arc4-1
4
1
0.020408
11.05
}
{
Arc4-2
4
2
1
3.63
}
{
Arc4-3
4
3
0.025641
4.34
}
{
Arc4-5

4
5
0.014706
6.85
}
{
Arc5-1
5
1
0.011111
4.72
}
{
Arc5-2
5
2
1
6.20
}
{
Arc5-3
5
3
0.025641
7.60
}
{
Arc5-4
5
4
0.037037
6.85
}
DEMANDS
{
Dem2-1
2
0
1440
0
1
0
1440
0
1439
}
{
Dem3-1
3
0
1440
0
1
0
1440
0
1439
}
{
Dem4-1
4
0
1440
0
1
0
1440
0
1439
}
{
Dem5-1
5
0
1440
0
1
0
1440
0
1439
}

ΠΙΝΑΚΑΣ 13: Input AMPL

```
set VEHICLES := 1 ;

set VSD := vsd1 ;

set VED := ved1 ;

param Q :=
1 2
;

param MVRT:=
1 1440
;
param T:= 50;

param NtPickup := 4;

set Pplus := 1 2 3 4 ;

set Pminus := 5 6 7 8 ;

param earlier :=
1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
vsd1 0
ved1 0
;

param later :=
1 1440
2 1440
3 1440
4 1440
5 1440
6 1440
7 1440
8 1440
vsd1 1440
ved1 1440
;
param drt :=
```

```
1 1440
2 1440
3 1440
4 1440
5 1440
6 1440
7 1440
8 1440
;
param travel_time :
      vsd1  1     2     3     4     5     6     7     8     ved1  :=
vsd1  0     9.14  10.24 11.04 4.71  0     0     0     0     0
1     9.14  0     1.43  3.63  6.20  9.14  9.14  9.14  9.14  9.14
2     10.24 1.4   0     4.33  7.60  10.24 10.24 10.24 10.24 10.24
3     11.04 3.63  4.33  0     6.85  11.04 11.04 11.04 11.04 11.04
4     4.71  6.20  7.60  6.85  0     4.71  4.71  4.71  4.71  4.71
5     0     9.14  10.24 11.04 4.71  0     0     0     0     0
6     0     9.14  10.24 11.04 4.71  0     0     0     0     0
7     0     9.14  10.24 11.04 4.71  0     0     0     0     0
8     0     9.14  10.24 11.04 4.71  0     0     0     0     0
ved1  0     9.14  10.24 11.04 4 71  0     0     0     0     0
;
param travel_cost :
      vsd1  1     2     3     4     5     6     7     8     ved1  :=
vsd1  0     0.043 0.016 0.020 0.011 0     0     0     0     0
1     0.043 0     0.026 0.038 0.014 0.043 0.043 0.043 0.043 0.043
2     0.016 1.000 0     0.037 0.014 0.016 0.016 0.016 0.016 0.016
3     0.020 1.000 0.025 0     0.014 0.020 0.020 0.020 0.020 0.020
4     0.011 1.000 0.025 0.037 0     0.011 0.011 0.011 0.011 0.011
5     0     0.043 0.016 0.020 0.011 0     0     0     0     0
6     0     0.043 0.016 0.020 0.011 0     0     0     0     0
7     0     0.043 0.016 0.020 0.011 0     0     0     0     0
8     0     0.043 0.016 0.020 0.011 0     0     0     0     0
ved1  0     0.043 0.016 0.020 0.011 0     0     0     0     0
;
param service_time :=
vsd1 0
1 1
2 1
3 1
4 1
5 1
6 1
7 1
8 1
ved1 0
;

param load :=
1 1
```

```
2 1
3 1
4 1
5 -1
6 -1
7 -1
8 -1
;
```

ΠΙΝΑΚΑΣ 14: Αποτελέσματα - Output Απλού ευρετικού αλγόριθμου.

Route ID :1	BACK :0
Vehicle ID :1	--------------
Route ID :1	FN :4
Vehicle ID :MONK1	TN :5
Route Cost :0.132412	Demand :Dem5-1
Route Time :43.44	Ei :0
Total Passengers Distance :0.09603	Li :1440
Average Passengers per Distance Unit	Mode :1
:0.85738	AAT :17.9
**************************************	ADT :17.9
Route Slope :0.0132412	WT :0
--------------	BACK :0
FN :1	--------------
TN :1	FN :5
Demand :SDEPOT	TN :1
Ei :0	Demand :Dem5-1
Li :1440	Ei :0
AAT :0	Li :1440
ADT :0	Mode :-1
WT :0	AAT :22.62
BACK :0	ADT :22.62
--------------	WT :0
FN :1	BACK :0
TN :4	--------------
Demand :Dem4-1	FN :1
Ei :0	TN :1
Li :1440	Demand :Dem4-1
Mode :1	Ei :0
AAT :11.05	Li :1440
ADT :11.05	Mode :-1
WT :0	AAT :22.62

ADT :22.62 WT :0 BACK :0 --------------- FN :1 TN :2 Demand :Dem2-1 Ei :0 Li :1440 Mode :1 AAT :31.76 ADT :31.76 WT :0 BACK :0 --------------- FN :2 TN :3 Demand :Dem3-1 Ei :0 Li :1440 Mode :1 AAT :33.19 ADT :33.19 WT :0 BACK :0 --------------- FN :3 TN :1 Demand :Dem3-1 Ei :0 Li :1440 Mode :-1	AAT :43.44 ADT :43.44 WT :0 BACK :0 --------------- FN :1 TN :1 Demand :Dem2-1 Ei :0 Li :1440 Mode :-1 AAT :43.44 ADT :43.44 WT :0 BACK :0 --------------- FN :1 TN :1 Demand :EDEPOT Ei :0 Li :1440 AAT :43.44 ADT :43.44 WT :0 BACK :0 STATISTICS Total Cost :0.132 Total Time :43.44 Totaıl Passenger Kilometers :0.09603	Average Passenger per Kilometer :0.85738 Total Demand SP Cost :0.09139 Total Demand real Cost :0.09603 Total Demand SP Time :35.16 Total Demand real Time :38.22 Average Time Deviation :0.765 Average Cost Deviation :0.00116 Average Cost Deviation Percentage :5 Average Time Deviation Percentage :8 Total Proccesing Time (Sec) :0.441

ΠΑΡΑΡΤΗΜΑ [Ε]

Ένας μοναχός – πέντε αντικείμενα

ΠΙΝΑΚΑΣ 15 : Αποτελέσματα **Matlab (calc)**

Από κόμβο:	Προς κόμβο:	Χρόνος:	1/Κόστος:	Κόστος:
1	2	8.14	0.028571	35
1	3	8.37	0.016393	61
1	4	5.95	0.017857	56
1	5	11.26	0.011111	90
1	6	12.13	0.010526	95
2	1	8.14	0.028571	35
2	3	7.45	0.038462	26
2	4	3.54	0.047619	21
2	5	3.44	0.018182	55
2	6	4.85	0.016667	60
3	1	8.37	0.016393	61
3	2	7.45	1	1
3	4	4.44	0.045455	22
3	5	7.96	0.017857	56
3	6	7.51	0.016393	61
4	1	5.95	0.017857	56
4	2	3.54	1	1
4	3	4.44	0.037037	27
4	5	5.69	0.017857	56

4	6	6.27	0.016393	61
5	1	11.26	0.011111	90
5	2	3.44	1	1
5	3	7.96	0.037037	27
5	4	5.69	0.045455	22
5	6	1.71	0.016393	61
6	1	12.13	0.010526	95
6	2	4.85	1	1
6	3	7.51	0.037037	27
6	4	6.27	0.045455	22
6	5	1.71	0.017857	56

ΠΙΝΑΚΑΣ 16: Αποτελέσματα - Output Απλού ευρετικού αλγόριθμου.

Route ID :1 ***Vehicle ID :1*** ***Route ID :1*** ***Vehicle ID :Vehicle1*** ***Route Cost :0.109317*** ***Route Time :43.86*** ***Total Passengers Distance :0.107268*** ***Average Passengers per Distance Unit :1.17284*** ********************************* ***Route Slope :0.0109317*** --------------- ***FN :1*** ***TN :1*** ***Demand :SDEPOT*** ***Ei :0*** ***Li :1440*** ***AAT :0*** ***ADT :0*** ***WT :0*** ***BACK :0*** --------------- ***FN :1*** ***TN :4*** ***Demand :DEM4-1*** ***Ei :0*** ***Li :1440*** ***Mode :1*** ***AAT :5.95*** ***ADT :5.95*** ***WT :0***	***BACK :0*** --------------- ***FN :4*** ***TN :3*** ***Demand :DEM3-1*** ***Ei :0*** ***Li :1440*** ***Mode :1*** ***AAT :10.39*** ***ADT :10.39*** ***WT :0*** ***BACK :0*** --------------- ***FN :3*** ***TN :1*** ***Demand :DEM4-1*** ***Ei :0*** ***Li :1440*** ***Mode :-1*** ***AAT :18.76*** ***ADT :18.76*** ***WT :0*** ***BACK :0*** --------------- ***FN :1*** ***TN :1*** ***Demand :DEM3-1*** ***Ei :0*** ***Li :1440*** ***Mode :-1*** ***AAT :18.76***

ADT :18.76 *WT :0* *BACK :0* --------------- *FN :1* *TN :5* *Demand :DEM5-1* *Ei :0* *Li :1440* *Mode :1* *AAT :30.02* *ADT :30.02* *WT :0* *BACK :0* --------------- *FN :5* *TN :6* *Demand :DEM6-1* *Ei :0* *Li :1440* *Mode :1* *AAT :31.73* *ADT :31.73* *WT :0* *BACK :0* --------------- *FN :6* *TN :1* *Demand :DEM5-1* *Ei :0* *Li :1440* *Mode :-1* *AAT :43.86*	*ADT :43.86* *WT :0* *BACK :0* --------------- *FN :1* *TN :1* *Demand :DEM6-1* *Ei :0* *Li :1440* *Mode :-1* *AAT :43.86* *ADT :43.86* *WT :0* *BACK :0* --------------- *FN :1* *TN :1* *Demand :EDEPOT* *Ei :0* *Li :1440* *AAT :43.86* *ADT :43.86* *WT :0* *BACK :0* *STATISTICS* *Total Cost :0.109317* *Total Time :43.86* *Total Passenger Kilometers :0.107268* *Average Passenger per Kilometer :1.17284* *Total Demand SP Cost :0.055887*	*Total Demand real Cost :0.107268* *Total Demand SP Time :37.71* *Total Demand real Time :47.15* *Average Time Deviation :2.36* *Average Cost Deviation :0.0128452* *Average Cost Deviation Percentage :91* *Average Time Deviation Percentage :25* *Total Proccesing Time (Sec) :0.451*

ΠΑΡΑΡΤΗΜΑ [Ζ]

Τρεις μοναχοί – δέκα αντικείμενα

ΠΙΝΑΚΑΣ 17: Αποτελέσματα **Matlab (calc)**

Από κόμβο:	Προς κόμβο:	Χρόνος:	1/Κόστος:	Κόστος:
1	2	11.29	0.043478	23
1	3	7.53	0.021739	46
1	4	2.62	0.020408	49
1	5	4.47	0.018519	54
1	6	11.15	0.013158	76
1	7	9.59	0.016393	61
1	8	4.56	0.017241	58
1	9	9.52	0.012195	82
1	10	7.52	0.011111	90
1	11	5.62	0.012821	78
2	1	11.29	0.043478	23
2	3	5.35	0.043478	23
2	4	8.78	0.038462	26
2	5	6.89	0.032258	31
2	6	0.41	0.018868	53
2	7	3.70	0.026316	38
2	8	6.81	0.028571	35
2	9	3.31	0.016949	59
2	10	8.87	0.014925	67

2	11	6.14	0.018182	55
3	1	7.53	0.021739	46
3	2	5.35	1	1
3	4	5.73	0.037037	27
3	5	3.45	0.03125	32
3	6	5.02	0.018519	54
3	7	2.22	0.025641	39
3	8	3.35	0.027778	36
3	9	2.33	0.016667	60
3	10	9.21	0.014706	68
3	11	4.92	0.017857	56
4	1	2.62	0.020408	49
4	2	8.78	1	1
4	3	5.73	0.041667	24
4	5	2.30	0.03125	32
4	6	8.68	0.018519	54
4	7	7.53	0.025641	39
4	8	2.40	0.027778	36
4	9	7.38	0.016667	60
4	10	5.63	0.014706	68
4	11	3.00	0.017857	56
5	1	4.47	0.018519	54
5	2	6.89	1	1
5	3	3.45	0.041667	24

5	4	2.30	0.037037	27
5	6	6.72	0.018519	54
5	7	5.25	0.025641	39
5	8	0.10	0.027778	36
5	9	5.12	0.016667	60
5	10	6.52	0.014706	68
5	11	2.45	0.017857	56
6	1	11.15	0.013158	76
6	2	0.41	1	1
6	3	5.02	0.041667	24
6	4	8.68	0.037037	27
6	5	6.72	0.03125	32
6	7	3.31	0.025641	39
6	8	6.64	0.027778	36
6	9	2.93	0.016667	60
6	10	9.03	0.014706	68
6	11	6.11	0.017857	56
7	1	9.59	0.016393	61
7	2	3.70	1	1
7	3	2.22	0.041667	24
7	4	7.53	0.037037	27
7	5	5.25	0.03125	32
7	6	3.31	0.018519	54
7	8	5.15	0.027778	36

7	9	0.45	0.016667	60
7	10	9.94	0.014706	68
7	11	5.95	0.017857	56
8	1	4.56	0.017241	58
8	2	6.81	1	1
8	3	3.35	0.041667	24
8	4	2.40	0.037037	27
8	5	0.10	0.03125	32
8	6	6.64	0.018519	54
8	7	5.15	0.025641	39
8	9	5.02	0.016667	60
8	10	6.57	0.014706	68
8	11	2.47	0.017857	56
9	1	9.52	0.012195	82
9	2	3.31	1	1
9	3	2.33	0.041667	24
9	4	7.38	0.037037	27
9	5	5.12	0.03125	32
9	6	2.93	0.018519	54
9	7	0.45	0.025641	39
9	8	5.02	0.027778	36
9	10	9.57	0.014706	68
9	11	5.65	0.017857	56
10	1	7.52	0.011111	90

10	2	8.87	1	1
10	3	9.21	0.041667	24
10	4	5.63	0.037037	27
10	5	6.52	0.03125	32
10	6	9.03	0.018519	54
10	7	9.94	0.025641	39
10	8	6.57	0.027778	36
10	9	9.57	0.016667	60
10	11	4.31	0.017857	56
11	1	5.62	0.012821	78
11	2	6.14	1	1
11	3	4.92	0.041667	24
11	4	3.00	0.037037	27
11	5	2.45	0.03125	32
11	6	6.11	0.018519	54
11	7	5.95	0.025641	39
11	8	2.47	0.027778	36
11	9	5.65	0.016667	60
11	10	4.31	0.014706	68

ΠΙΝΑΚΑΣ 18: Αποτελέσματα - Output Απλού ευρετικού αλγόριθμου.

Route ID :1 *Vehicle ID :1* *Route ID :1* *Vehicle ID :Vehicle1* *Route Cost :0.038012* *Route Time :26.61* *Total Passengers Distance :0.036928* *Average Passengers per Distance Unit* *:1.43038* ************************************* *Route Slope :0.00633533* --------------- *FN :1* *TN :1* *Demand :SDEPOT* *Ei :0* *Li :1440* *AAT :0* *ADT :0* *WT :0* *BACK :0* --------------- *FN :1* *TN :9* *Demand :DEM9-1* *Ei :0* *Li :1440* *Mode :1* *AAT :9.52* *ADT :9.52* *WT :0*	*BACK :0* --------------- *FN :9* *TN :10* *Demand :DEM10-1* *Ei :0* *Li :1440* *Mode :1* *AAT :19.09* *ADT :19.09* *WT :0* *BACK :0* --------------- *FN :10* *TN :1* *Demand :DEM9-1* *Ei :0* *Li :1440* *Mode :-1* *AAT :26.61* *ADT :26.61* *WT :0* *BACK :0* --------------- *FN :1* *TN :1* *Demand :DEM10-1* *Ei :0* *Li :1440* *Mode :-1* *AAT :26.61*

ADT :26.61

WT :0

BACK :0

FN :1

TN :1

Demand :EDEPOT

Ei :0

Li :1440

AAT :26.61

ADT :26.61

WT :0

BACK :0

****

Route ID :2

Vehicle ID :2

Route ID :2

Vehicle ID :Vehicle2

Route Cost :0.043836

Route Time :22.88

Total Passengers Distance :0.043499

Average Passengers per Distance Unit :1.41792

****

Route Slope :0.007306

FN :1

TN :1

Demand :SDEPOT

Ei :0

Li :1440

AAT :0

ADT :0

WT :0

BACK :0

FN :1

TN :6

Demand :DEM6-1

Ei :0

Li :1440

Mode :1

AAT :11.15

ADT :11.15

WT :0

BACK :0

FN :6

TN :11

Demand :DEM11-1

Ei :0

Li :1440

Mode :1

AAT :17.26

ADT :17.26

WT :0

BACK :0

FN :11

TN :1

Demand :DEM6-1

Ei :0

Li :1440

Mode :-1

AAT :22.88

ADT :22.88

WT :0 ***BACK :0*** --------------- ***FN :1*** ***TN :1*** ***Demand :DEM11-1*** ***Ei :0*** ***Li :1440*** ***Mode :-1*** ***AAT :22.88*** ***ADT :22.88*** ***WT :0*** ***BACK :0*** --------------- ***FN :1*** ***TN :1*** ***Demand :EDEPOT*** ***Ei :0*** ***Li :1440*** ***AAT :22.88*** ***ADT :22.88*** ***WT :0*** ***BACK :0*** ** ***Route ID :3*** ***Vehicle ID :3*** ***Route ID :3*** ***Vehicle ID :Vehicle3*** ***Route Cost :0.129452*** ***Route Time :28.69*** ***Total Passengers Distance :0.126715*** ***Average Passengers per Distance Unit :1.16205*** **	***Route Slope :0.0129452*** --------------- ***FN :1*** ***TN :1*** ***Demand :SDEPOT*** ***Ei :0*** ***Li :1440*** ***AAT :0*** ***ADT :0*** ***WT :0*** ***BACK :0*** --------------- ***FN :1*** ***TN :4*** ***Demand :DEM4-1*** ***Ei :0*** ***Li :1440*** ***Mode :1*** ***AAT :2.62*** ***ADT :2.62*** ***WT :0*** ***BACK :0*** --------------- ***FN :4*** ***TN :5*** ***Demand :DEM5-1*** ***Ei :0*** ***Li :1440*** ***Mode :1*** ***AAT :4.92*** ***ADT :4.92*** ***WT :0*** ***BACK :0***

--------------- *FN :5* *TN :1* *Demand :DEM4-1* *Ei :0* *Li :1440* *Mode :-1* *AAT :9.39* *ADT :9.39* *WT :0* *BACK :0* --------------- *FN :1* *TN :1* *Demand :DEM5-1* *Ei :0* *Li :1440* *Mode :-1* *AAT :9.39* *ADT :9.39* *WT :0* *BACK :0* --------------- *FN :1* *TN :8* *Demand :DEM8-1* *Ei :0* *Li :1440* *Mode :1* *AAT :13.95* *ADT :13.95* *WT :0* *BACK :0*	--------------- *FN :8* *TN :7* *Demand :DEM7-1* *Ei :0* *Li :1440* *Mode :1* *AAT :19.1* *ADT :19.1* *WT :0* *BACK :0* --------------- *FN :7* *TN :1* *Demand :DEM8-1* *Ei :0* *Li :1440* *Mode :-1* *AAT :28.69* *ADT :28.69* *WT :0* *BACK :0* --------------- *FN :1* *TN :1* *Demand :DEM7-1* *Ei :0* *Li :1440* *Mode :-1* *AAT :28.69* *ADT :28.69* *WT :0* *BACK :0*	--------------- *FN :1* *TN :1* *Demand :EDEPOT* *Ei :0* *Li :1440* *AAT :28.69* *ADT :28.69* *WT :0* *BACK :0* *STATISTICS* *Total Cost :0.2113* *Total Time :78.18* *Total Passenger Kilometers :0.207142* *Average Passenger per Kilometer :1.25132* *Total Demand SP Cost :0.121846* *Total Demand real Cost :0.207142* *Total Demand SP Time :55.05* *Total Demand real Time :77.53* *Average Time Deviation :2.81* *Average Cost Deviation :0.010662* *Average Cost Deviation Percentage :70* *Average Time Deviation Percentage :40* *Total Proccesing Time (Sec) :1.07*