



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

Επίλυση του Προβλήματος Δρομολόγησης Οχημάτων Χωρίς
Επιστροφή στην Αποθήκη με τον
Αλγόριθμο Βελτιστοποίησης Αποικίας Μυρμηγκιών

Συγγραφέας:
ΣΠΥΡΙΔΩΝ ΑΥΡΗΛΙΩΝΗΣ

Επιβλέπων Καθηγητής:
ΔΡ. ΜΑΡΙΝΑΚΗΣ ΙΩΑΝΝΗΣ

ΝΟΕΜΒΡΙΟΣ, 2010

Ευχαριστίες

Δράττομαι της ευκαιρίας για να ευχαριστήσω προσωπικά τον επιβλέποντα καθηγητή μου κύριο Μαρινάκη Ιωάννη, που μου έδωσε την ευκαιρία να εργασθώ σε ένα τόσο ενδιαφέρον αλλά και δυναμικά εξελισσόμενο ερευνητικό πεδίο.

Ακολούθως θα ήθελα να αφιερώσω αυτή την διπλωματική εργασία στην οικογένεια μου και ιδιαίτερα στην μητέρα μου Παναγιώτα ως ελάχιστη ένδειξη ευγνωμοσύνης για την αμέριστη συμπαράσταση που μου παρείχε καθόλη τη διάρκεια των σπουδών μου στο Πολυτεχνείο Κρήτης.

Τέλος, ευχαριστώ όλους μου τους φίλους για την ψυχολογική υποστήριξη που μου παρείχαν συνεχώς όλα αυτά τα χρόνια. Ιδιαίτερη μνεία αξίζει στον Φίλο και μεταπτυχιακό φοιτητή του τμήματος Ηλεκτρονικών Μηχανικών και Μηχανικών Υπολογιστών, Μαρκόπουλο Παναγιώτη, ο οποίος με κατέστησε επιδέξιο χρήστη του λογισμικού Matlab.

Περιεχόμενα

<u>Κεφάλαιο 1: Εισαγωγή</u>	7
1.1. Logistics και Εφοδιαστική Αλυσίδα.....	7
1.2. Πλαίσιο εργασίας των Logistics.....	8
<u>Κεφάλαιο 2: Προβλήματα δρομολόγησης οχημάτων</u>	9
2.1. Εισαγωγή στο πρόβλημα δρομολόγησης οχημάτων.....	9
2.2. Μαθηματική Προτυποποίηση.....	11
2.3. Επεκτάσεις απλού VRP.....	12
2.4. Πρόβλημα Δρομολόγησης Οχημάτων, χωρίς επιστροφή στην αποθήκη – OVRP (Open Vehicle Routing Problem).....	13
<u>Κεφάλαιο 3: Βελτιστοποίηση σε Προβλήματα Δρομολόγησης Οχημάτων</u>	15
3.1. Αλγόριθμοι βελτιστοποίησης προβλημάτων της εφοδιαστικής αλυσίδας.....	15
3.2. Ευρετικοί αλγόριθμοι που εφαρμόζονται στη διπλωματική εργασία.....	17
3.2.1. Αλγόριθμος πλησιέστερου γείτονα.....	17
3.2.2. Αλγόριθμος 2-OPT.....	17
3.3. Αλγόριθμος βελτιστοποίησης αποικίας μυρμηγκιών (Ant Colony Optimization –ACO).....	19
3.3.1. Αλγόριθμος βελτιστοποίησης αποικίας μυρμηγκιών στο πρόβλημα του πλανόδιου πωλητή – TSP (Travelling Salesman Problem).....	20
<u>Κεφάλαιο 4: Εφαρμογή ACO στο OVRP</u>	22
4.1. Παρουσίαση του προβλήματος προς βελτιστοποίηση.....	22
4.2. Βέλτιστες λύσεις των παραδειγμάτων σε παγκόσμια κλίμακα.....	24
4.3. Αλγοριθμική προσέγγιση προβλήματος.....	25
<u>Κεφάλαιο 5: Αποτελέσματα ACO στο OVRP</u>	27
5.1. Παράδειγμα C1.....	27
5.1.1. Αποτελέσματα εφαρμογής ACO στο παράδειγμα 1.....	27
5.1.2. Γραφική απεικόνιση λύσης παραδείγματος 1.....	28
5.2. Παράδειγμα C2.....	29
5.2.1. Αποτελέσματα εφαρμογής ACO στο παράδειγμα 2.....	29
5.2.2. Γραφική απεικόνιση λύσης παραδείγματος 2.....	30
5.3. Παράδειγμα C3.....	31
5.3.1. Αποτελέσματα εφαρμογής ACO στο παράδειγμα 3.....	31
5.3.2. Γραφική απεικόνιση λύσης παραδείγματος 3.....	32
5.4. Παράδειγμα C4.....	33
5.4.1. Αποτελέσματα εφαρμογής ACO στο παράδειγμα 4.....	33
5.4.2. Γραφική απεικόνιση λύσης παραδείγματος 4.....	34
5.5. Παράδειγμα C5.....	35
5.5.1. Αποτελέσματα εφαρμογής ACO στο παράδειγμα 5.....	35
5.5.2. Γραφική απεικόνιση λύσης παραδείγματος 5.....	36
5.6. Παράδειγμα C6.....	37
5.6.1. Αποτελέσματα εφαρμογής ACO στο παράδειγμα 6.....	37
5.6.2. Γραφική απεικόνιση λύσης παραδείγματος 6.....	38

5.7. Παράδειγμα C7.....	39
5.7.1. Αποτελέσματα εφαρμογής ACO στο παράδειγμα 7.....	39
5.7.2. Γραφική απεικόνιση λύσης παραδείγματος 7.....	40
5.8. Παράδειγμα C8.....	41
5.8.1. Αποτελέσματα εφαρμογής ACO στο παράδειγμα 8.....	41
5.8.2. Γραφική απεικόνιση λύσης παραδείγματος 8.....	42
5.9. Παράδειγμα C9.....	43
5.9.1. Αποτελέσματα εφαρμογής ACO στο παράδειγμα 9.....	43
5.9.2. Γραφική απεικόνιση λύσης παραδείγματος 9.....	44
5.10. Παράδειγμα C10.....	45
5.10.1. Αποτελέσματα εφαρμογής ACO στο παράδειγμα 10.....	45
5.10.2. Γραφική απεικόνιση λύσης παραδείγματος 10.....	46
5.11. Παράδειγμα C11.....	47
5.11.1. Αποτελέσματα εφαρμογής ACO στο παράδειγμα 11.....	47
5.11.2. Γραφική απεικόνιση λύσης παραδείγματος 11.....	48
5.12. Παράδειγμα C12.....	49
5.12.1. Αποτελέσματα εφαρμογής ACO στο παράδειγμα 12.....	49
5.12.2. Γραφική απεικόνιση λύσης παραδείγματος 12.....	50
5.13. Παράδειγμα C13.....	51
5.13.1. Αποτελέσματα εφαρμογής ACO στο παράδειγμα 13.....	51
5.13.2. Γραφική απεικόνιση λύσης παραδείγματος 13.....	52
5.14. Παράδειγμα C14.....	53
5.14.1. Αποτελέσματα εφαρμογής ACO στο παράδειγμα 14.....	53
5.14.2. Γραφική απεικόνιση λύσης παραδείγματος 14.....	54
 <u>Κεφάλαιο 6: Σύνοψη – Σχόλια</u>	55
6.1. Συνολική παρουσίαση αποτελεσμάτων.....	55
6.2. Σχολιασμός.....	56
 <u>Παράρτημα</u>	57
 <u>Βιβλιογραφία</u>	89

Κεφάλαιο 1 – Εισαγωγή

1.1. Logistics και Εφοδιαστική Αλυσίδα

Η έννοια των Logistics και του σχεδιασμού της εφοδιαστικής αλυσίδας κάθε άλλο παρά καινούριες μπορούν να θεωρηθούν. Από τη μεταφορά των πρώτων υλών για το χτίσιμο των πυραμίδων και της Ακρόπολης της Αθήνας, μέχρι τον ανεφοδιασμό των αρχαιοτάτων στρατιών σπουδαίο ρόλο έπαιζε ο τρόπος με τον οποίο θα γινόταν δυνατή η ροή των υλικών από τον προμηθευτή ή τον κατασκευαστή μέχρι τον καταναλωτή ή ενδιαφερόμενο, καθώς και η πιθανή ροή πληροφοριών μεταξύ των μελών αυτής της αλυσίδας.

Στις αρχές του 1900 αναγνωρίστηκαν τα συγκριτικά πλεονεκτήματα που θα μπορούσε να προσφέρει η χρήση των Logistics στον ανταγωνισμό των μεγάλων επιχειρήσεων. Έτσι άρχισαν οι προσπάθειες για την βελτιστοποίηση της παραγωγής-εξυπηρέτησης ως σύνολο και όχι ως χωριστές μονάδες της παραγωγικής διαδικασίας. Σύμφωνα με το [2], ήταν ο Arch Shawo οποίος διατύπωσε πως είναι μεγαλύτερης σημασίας η βελτιστοποίηση της εφοδιαστικής αλυσίδας σαν σύνολο από ότι σε μεμονωμένα κομμάτια.

Τα logistics αποσκοπούν στην παραγωγή προϊόντων με όσο το δυνατόν χαμηλότερο κόστος, στη διατήρηση των προϊόντων με τον καλύτερο δυνατό τρόπο, στην πλήρη αξιοποίηση των υλικών μέσων της επιχείρησης, στη μεταφορά των προϊόντων με το χαμηλότερο δυνατό κόστος και τις μικρότερες δυνατές καθυστερήσεις και τελικά στην επίτευξη κερδοφορίας και οικονομίας για την επιχείρηση. Αν και αφορούν σε κάθε είδους επιχειρηματικό τομέα και κάθε είδους επιχείρηση, εκεί που βρίσκουν κατεξοχήν πρόσφορο έδαφος εφαρμογής είναι οι επιχειρήσεις που δραστηριοποιούνται στην εφοδιαστική αλυσίδα (εμπορικές, μεταφορικές, παραγωγικές, εταιρίες αποθήκευσης κ.ά.), ανεξαρτήτως του τομέα της οικονομίας στον οποίο ανήκουν. Ο όρος "εφοδιαστική αλυσίδα" περιγράφει το πλέγμα διαδικασιών που απαιτούνται ώστε ένα προϊόν να περάσει από τη φάση της παραγωγής στη φάση της κατανάλωσης. Μεταξύ των διαδικασιών αυτών ξεχωρίζει η παραγωγή, η τυποποίηση, η αποθήκευση, η διακίνηση και η διάθεση του προϊόντος.

Όπως διαφαίνεται από τα παραπάνω, logistics και εφοδιαστική αλυσίδα συνδέονται άρρηκτα. Η εφοδιαστική αλυσίδα αποτελεί το βασικότερο πεδίο εφαρμογής των logistics, τα οποία αποτελούν το βασικότερο ζητούμενο για την ορθολογικοποίηση και την επιτυχία των διαδικασιών της εφοδιαστικής αλυσίδας. Τα logistics απαντούν στο πώς πρέπει να οργανωθούν οι διαδικασίες της εφοδιαστικής αλυσίδας.

1.2. Πλαίσιο εργασίας των Logistics

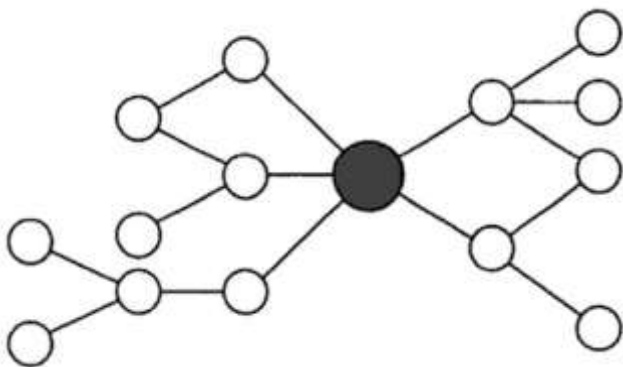
Οι βασικοί τομείς ενασχόλησης των logistics είναι επιγραμματικά [1]:

1. Εξυπηρέτηση Πελατών (Customer Service)
2. Ανταλλακτικά και Υπηρεσίες μετά την Πώληση (Parts and Service Support)
3. Συσκευασία (Packaging)
4. Δημοσιεύσεις-Υποστήριξη του Προϊόντος (Publications)
5. Τροφοδοσία (Procurement)
6. Διαχείριση Υλικών (Material Handling)
7. Διαχείριση Αποθηκών και Κέντρων Διανομών (Warehousing and Storage)
8. Διαχείριση Επισωρευθέντων Προϊόντων (Return Goods Handling)
9. Αντίστροφη Εφοδιαστική (Reverse Logistics)
10. Πρόβλεψη Ζήτησης (Demand Forecasting)
11. Χωροθέτηση Εγκαταστάσεων (Plant and Ware house Location)
12. Διαχείριση Αποθεμάτων (Inventory Management)
13. **Μεταφορές και Διανομή (Transportation and Distribution)**
14. Επικοινωνία και Τεχνολογία Επικοινωνίας (Communication and Communication Technology)
15. Επεξεργασία Παραγγελιών και Τεχνολογία Πληροφορικής (Other Processing and Information Technology)
16. Εκπαίδευση Προσωπικού (Personnel Education)
17. Εφοδιαστική Τρίτου (Third Party Logistics) και Στρατηγικός Συνεταιρισμός (Strategic Partnering)
18. Συγκέντρωση και Κεντροποίηση (Concentration and Centralization)
19. Παγκοσμιοποιημένη Εφοδιαστική (Global Logistics)

Στην παρούσα διπλωματική εργασία θα εξετασθεί το πρόβλημα της μεταφοράς και διανομής προϊόντων μεταξύ του χώρου αποθήκευσης και των τελικών προορισμών-πελατών. Το πρόβλημα αυτό είναι γνωστό ως πρόβλημα δρομολόγησης οχημάτων ή Vehicle Routing Problem, στην παγκόσμια βιβλιογραφία. Ποιο συγκεκριμένα θα αναλυθεί το Open Vehicle Routing Problem, του οποίου επεξήγηση θα ακολουθήσει σε επόμενο κεφάλαιο. Στη συνέχεια της εργασίας, για λόγους συντομίας, θα χρησιμοποιείται ο όρος VRP και OVRP για το κάθε πρόβλημα αντίστοιχα.

2.1. Εισαγωγή στο πρόβλημα δρομολόγησης οχημάτων

Ένα σύστημα μεταφοράς μπορεί να αναπαρασταθεί με τη μορφή δικτύου κόμβων και τόξων, όπου οι κόμβοι αντιπροσωπεύουν τοποθεσίες και τα τόξα διαδρομές ή συνδέσμους μεταξύ αυτών.



Στη βασική περίπτωση του VRP ένα όχημα ξεκινά από την αποθήκη και εξυπηρετεί πελάτες με προκαθορισμένη σειρά επίσκεψης, επιστρέφοντας στην αποθήκη για επαναφόρτωση όταν αυτό κρίνεται σκόπιμο. Στόχος είναι η εξυπηρέτηση όλων των πελατών και η ελαχιστοποίηση της διανυθείσας απόστασης (κόστους). Σε ειδικές περιπτώσεις είναι σκόπιμη η ελαχιστοποίηση των ποινών-κυρώσεων που προκύπτουν από τη μερική εξυπηρέτηση των πελατών ή η εξισορρόπηση των διαδρομών των οχημάτων, με βάση το χρόνο μετάβασης και το φορτίο μεταφοράς.

Στην κλασική εκδοχή του VRP, δημιουργείται μία συχνότητα από εξυπηρετήσεις για κάθε όχημα ομογενούς στόλου, βασισμένη σε μία μόνο αποθήκη. Έτσι ο σκοπός είναι να εξυπηρετηθούν όλοι οι πελάτες ελαχιστοποιώντας την διανυθείσα από τα οχήματα απόσταση. Κάθε όχημα έχει σταθερή χωρητικότητα και ίσως ένα περιορισμό δρομολογίων ο οποίος οριοθετεί την απόσταση που μπορεί να διανύσει. Επίσης κάθε πελάτης έχει γνωστή ζήτηση και εξυπηρετείται από ένα μόνο όχημα με μία μόνο επίσκεψη. Στο τέλος κάθε όχημα πρέπει να επιστέψει στην αποθήκη.

Γενικά η επίλυση ενός VRP δύναται να έχει τους κάτωθι στόχους προς βελτιστοποίηση:

- Μέγεθος στόλου οχημάτων
- Αριθμός οδηγών
- Αριθμός πραγματοποιήσιμων διαδρομών ανά ημέρα
- Μέσος αριθμός στάσεων σε κάθε διαδρομή
- Διαδρομές εντός και εκτός πόλης
- Συνολικό κόστος διανομής
- Κόστος πληρωμάτων
- Μελλοντικές απαιτήσεις, πρόβλεψη ζήτησης
- Υπολογιστική δύναμη της εταιρίας-οργανισμού για την υποστήριξη του δικτύου διανομής
- Συνδυασμός δρομολογίων με άλλες δραστηριότητες

2.2. Μαθηματική Προτυποποίηση

Το **πρόβλημα δρομολόγησης οχημάτων** είναι ένα πρόβλημα συνδυαστικής βελτιστοποίησης και κατά βάση ακεραίου προγραμματισμού, στο οποίο ένας αριθμός πελατών πρέπει να εξυπηρετηθεί από ένα στόλο οχημάτων. Παρουσιάστηκε για πρώτη φορά το 1959 από τους Dantzig και Ramser οι οποίοι την εποχή εκείνη περιέγραψαν έναν τρόπο για την βέλτιστη μεταφορά βενζίνης σε έναν αριθμό σταθμών τροφοδοσίας. Λίγα χρόνια αργότερα και συγκεκριμένα το 1964 οι Wright και Clarke πρότειναν έναν ευρετικό αλγόριθμο απληστίας ο οποίος ήταν σημαντικά βελτιωμένος σε σχέση με αυτόν των Danzig-Ramser.

Υπάρχουν αρκετές παραλλαγές και επεκτάσεις του κλασικού VRP. Παρακάτω παραθέεται η μοντελοποίηση του «Περιορισμένης Χωρητικότητας Πρόβλημα Δρομολόγησης Οχημάτων» (CVRP – Capacitated Vehicle Routing Problem) με σκοπό την παρουσίαση επιπλέον περιορισμών και πιο εξελιγμένων αντικειμενικών συναρτήσεων. Έτσι το βασικό πρόβλημα CVRP μπορεί να αναπαρασταθεί σύμφωνα με τα [1] και [3] ως:

Αντικειμενική Συνάρτηση:

$$\min \sum_{i,j} c_{i,j} \sum_k x_{ijk} \quad (2.2.1)$$

Υπό τους περιορισμούς:

$$\sum_k y_{ik} = \begin{cases} 1, & i = 2, \dots, n \\ m, & i = 1 \end{cases} \quad (2.2.2)$$

$$\sum_i q_i y_{ij} \leq Q_k, \quad k = 1, \dots, m \quad (2.2.3)$$

$$\sum_j x_{ijk} = \sum_j x_{jik} = y_{ik}, \quad \begin{cases} i = 1, \dots, n \\ k = 1, \dots, m \end{cases} \quad (2.2.4)$$

$$\sum_{i,j \in S} x_{ijk} \leq |S| - 1 \quad (2.2.5)$$

$$y_{ik} \in \{0, 1\}, \begin{cases} i = 1, \dots, n \\ k = 1, \dots, m \end{cases} \quad (2.2.6)$$

$$x_{ijk} \in \{0, 1\}, \begin{cases} k = 1, \dots, m \\ i, j = 1, \dots, n \end{cases} \quad (2.2.7)$$

Ο περιορισμός (2.2.1) εξασφαλίζει την εκχώρηση κάθε κόμβου-πελάτη μόνο σε ένα όχημα, εκτός του κόμβου-αποθήκης ο οποίος δέχεται όλα τα οχήματα. Ο (2.2.2) είναι ο περιορισμός χωρητικότητας των οχημάτων Q_k , όπου k ο αριθμός των οχημάτων. Ο (2.2.3) δηλώνει πως ένα όχημα που επισκέπτεται έναν πελάτη αναγκαστικά φεύγει από αυτόν. Επίσης για τα y_{ik} και x_{ijk} ισχύει $y_{ik}=1$, εάν το όχημα k επισκέπτεται τον πελάτη j αμέσως μετά τον i , αλλιώς είναι ίσο με μηδέν. Ανάλογα $x_{ijk}=1$, εάν ο πελάτης i εξυπηρετείται από το όχημα k , αλλιώς $x_{ijk}=0$. Οι παραπάνω συνθήκες ελήφθησαν και στο πρόβλημα της παρούσης εργασίας.

2.3. Επεκτάσεις απλού VRP

Αξίζει εδώ να αναφερθούν επιγραμματικά κάποιες σημαντικές επεκτάσεις του προβλήματος δρομολόγησης οχημάτων:

1. Περιορισμένης Χωρητικότητας Πρόβλημα Δρομολόγησης Οχημάτων (Capacitated Vehicle Routing Problem)
2. Πρόβλημα Δρομολόγησης Οχημάτων Με Παραλαβές (Vehicle Routing Problem with Backhauls)
3. Πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα (Vehicle Routing Problem with Time Windows)
4. Πρόβλημα Δρομολόγησης Οχημάτων με Παραλαβές και Διανομές κατά τη διάρκεια της Διαδρομής (Vehicle Routing Problem with Pickup and Delivery).
5. Πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Διαστήματα, Παραλαβές και Διανομές (Vehicle Routing Problem with Time Windows and Pickup and Delivery)
6. **Πρόβλημα Δρομολόγησης Οχημάτων, χωρίς επιστροφή στην αποθήκη (Open Vehicle Routing Problem-OVRP)**
7. Πρόβλημα Δρομολόγησης Οχημάτων με Χρονικά Διαστήματα, χωρίς επιστροφή στην αποθήκη (Open Vehicle Routing Problem with Time Windows)
8. Πρόβλημα Δρομολόγησης Οχημάτων με Δύο Είδη Πελατών κατά τη διάρκεια της Διαδρομής (Vehicle Routing Problem with backhauls and line-hauls customers)
9. Στοχαστικό Πρόβλημα Δρομολόγησης Οχημάτων (Stochastic or Probabilistic Vehicle Routing Problem)
10. Προγραμματισμός Οχημάτων (Vehicle Scheduling Problem – VSP)

2.4. Πρόβλημα Δρομολόγησης Οχημάτων, χωρίς επιστροφή στην αποθήκη – OVRP (Open Vehicle Routing Problem)

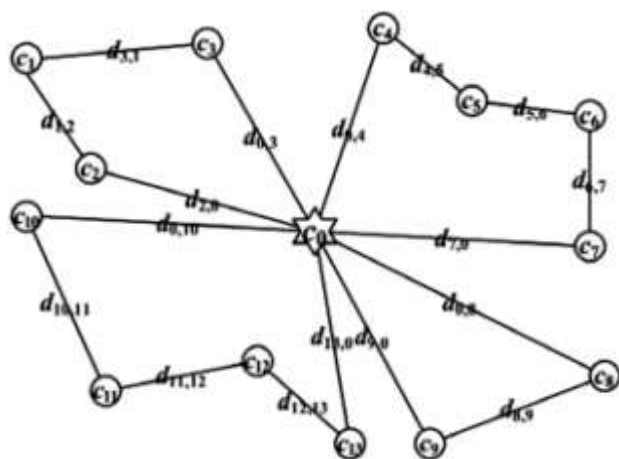
Η βασική διαφορά του OVRP με το VRP είναι πως σε αυτή την περίπτωση το όχημα που εξυπηρετεί τους πελάτες δεν επιστρέφει στην αποθήκη όταν ολοκληρώσει τις επισκέψεις σε όλους τους κόμβους-πελάτες. Αυτή η παραλλαγή του κλασικού VRP εμφανίστηκε στην βιβλιογραφία πάνω από είκοσι χρόνια πριν, αλλά μόλις πρόσφατα τράβηξε το ενδιαφέρον των ερευνητών και μελετητών. Σήμερα το OVRP εφαρμόζεται έμπρακτα κυρίως στη διανομή πακέτων και εφημερίδων. Μεταφορείς οι οποίοι δεν είναι εργαζόμενοι στην εταιρία παραδόσεων, χρησιμοποιούν τα δικά τους οχήματα για τις μεταφορές και συνεπώς στο τέλος των εξυπηρετήσεων δεν επιστρέφουν στην αποθήκη ή στην εταιρία. Εταιρίες όπως η FedEx [4] εφαρμόζουν στην πράξη αυτό το πρότυπο εξυπηρέτησης. Ένα αεροπλάνο απογειώνεται από ένα αεροδρόμιο-αποθήκη της εταιρίας, κάνει τις παραδόσεις και δεν επιστρέφει στην αποθήκη, παρά μένει στο τελευταίο αεροδρόμιο.

Σε αντίθεση με το VRP, εδώ το όχημα δεν διανύει ακριβή κύκλο αφού δεν επιστρέφει στην αποθήκη, δηλαδή οι κόμβοι συνδέονται με την αποθήκη μέσω ενός μονοπατιού Χάμιλτον (Hamiltonian Path). Επιπροσθέτως πρέπει να υπολογισθεί και ο ελάχιστος απαιτούμενος αριθμός οχημάτων που απαιτούνται για να εξυπηρετηθούν όλοι οι πελάτες.

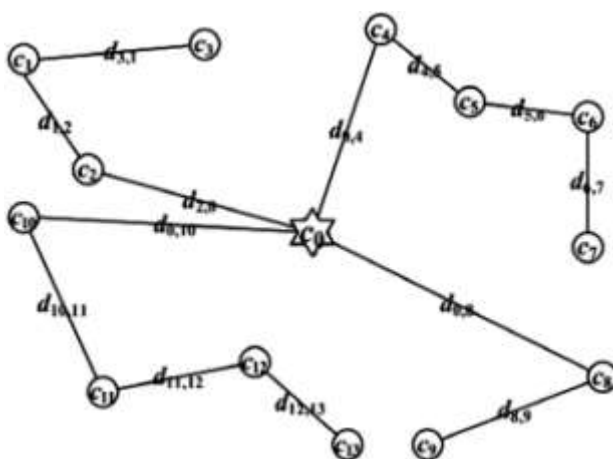
Τα τελευταία χρόνια μέθοδοι όπως tabu search, προσομοιωμένη απόπτωση, αλγόριθμος πλησιέστερου γείτονα, έχουν εφαρμοστεί για την επίλυση του συγκεκριμένου προβλήματος με επιτυχία.

Ένα θέμα που προκύπτει όταν ένα πρόβλημα δρομολόγησης οχημάτων ταιριάζει με το πλαίσιο εργασίας του OVRP είναι η μοντελοποίηση αυτού. Στην μοντελοποίηση και στην επίλυση πραγματικών προβλημάτων δρομολόγησης οχημάτων ο Levy [5] παρατήρησε ότι εάν οι εταιρίες δεν πλήρωναν μετά την τελευταία παράδοση τότε υπάρχει ένα σημαντικό κομμάτι διαδρομής το οποίο δεν σχετίζεται με την επιστροφή στην αποθήκη. Συχνά αυτό εφαρμόζεται στην πράξη όταν οι παραδόσεις γίνονται από εξωτερικούς συνεργάτες των εταιριών διανομής. Αυτοί έχουν τα δικά τους οχήματα και πληρώνουν οι ίδιοι το κόστος των οχημάτων και ίσως αποζημιώνονται με βάση την απόσταση που θα διανύσουν προς την παράδοση. Η εταιρία δεν χρειάζεται το όχημα πίσω στην βάση της.

Από τις αρχές του 1980 έως και τα τέλη του 1990 το OVRP δεν λάμβανε ιδιαίτερη σημασία στην βιβλιογραφία της επιχειρησιακής έρευνας. Ωστόσο από το 2000 αρκετοί ερευνητές χρησιμοποίησαν διάφορες μεθόδους για την επίλυση του.



Σχηματική απεικόνιση VRP



Σχηματική απεικόνιση OVRP

Κεφάλαιο 3 –Βελτιστοποίηση σε Προβλήματα Δρομολόγησης Οχημάτων

3.1. Αλγόριθμοι βελτιστοποίησης προβλημάτων της εφοδιαστικής αλυσίδας

Αξίζει να αναφερθούν επιγραμματικά οι σημαντικότερες από τις μεθοδολογίες για την επίλυση ενός τέτοιου προβλήματος. Αναφορικά λοιπόν μπορούν να κατηγοριοποιηθούν ως:

- Μαθηματικός προγραμματισμός
 - Γραμμικός προγραμματισμός
 - Αλγόριθμος Simplex
 - Ανάλυση ευαισθησίας
 - Ακέραιος και Μεικτός Ακέραιος Προγραμματισμός
 - Χαλάρωση του Lagrange
 - Δυναμική δημιουργία μεταβλητών
- Απλοί ευρετικοί αλγόριθμοι
 - Αλγόριθμοι απληστίας (greedy algorithms)
 - **Αλγόριθμος πλησιέστερου γείτονα**
 - Προσεγγιστικοί αλγόριθμοι (approximation algorithms)
 - Αλγόριθμοι τοπικής αναζήτησης (local search algorithms)
 - **2-OPT**
 - 3-OPT
 - k-OPT
 - SWAP
 - Path re-linking
 - WalkSAT και GSAT
- Μεθευρετικοί αλγόριθμοι
 - Προσομοιωμένη Ανόπτηση
 - Περιορισμένη Αναζήτηση
 - Γενετικοί και Εξελικτικοί Αλγόριθμοι
 - Αλγόριθμοι Νοημοσύνης Σμήνους
 - **Βελτιστοποίησης Αποικίας Μυρμηγκιών**
 - Βελτιστοποίησης Σμήνους Σωματιδίων
 - Βελτιστοποίησης Ζευγαρώματος Μελισσών
 - Αλγόριθμος Διασκορπισμένης Αναζήτησης
 - Διαδικασία Απληστης Τυχαιοποιημένης Προσαρμοστικής Αναζήτησης
 - Αλγόριθμος Διαφορικής Εξέλιξης

- Νευρωνικά Δίκτυα

Με έντονα γράμματα αναγράφονται οι αλγόριθμοι που χρησιμοποιήθηκαν στα πλαίσια αυτής της εργασίας.

Οι ευρετικοί αλγόριθμοι παράγουν λύσεις καλής ποιότητας σε σχετικά σύντομο χρονικό διάστημα. Αυτό συμβαίνει γιατί πραγματοποιούν περιορισμένη αναζήτηση στο διάστημα των πιθανών λύσεων. Βασικά χαρακτηριστικά των ευρετικών προσεγγίσεων είναι:

- Πρώτον, η αξιοποίηση χαρακτηριστικών του προβλήματος όσον αφορά τα κριτήρια και τη σειρά με την οποία επιλέγονται οι υποβέλτιστες λύσεις και επαληθεύονται οι περιορισμοί.
- Δεύτερον, το ενδεχόμενο στο ίδιο πρόβλημα αλλά για ορισμένες τιμές των παραμέτρων να παρέχει καλύτερες λύσεις κάποιος ευρετικός αλγόριθμος ενώ για άλλες τιμές άλλος .

Οι μεθευρετικοί αλγόριθμοι είναι μέθοδοι επίλυσης που συνδυάζουν διαδικασίες τοπικής αναζήτησης και υψηλότερου επιπέδου στρατηγικές για να δημιουργήσουν μια διαδικασία που είναι ικανή να ξεφύγει από κάποιο τοπικό ελάχιστο. Τα τελευταία χρόνια οι περισσότεροι αλγόριθμοι που έχουν αναπτυχθεί για την επίλυση των προβλημάτων συνδυαστικής βελτιστοποίησης ανήκουν σε αυτήν την κατηγορία. Οι μεθευρετικοί αλγόριθμοι συνήθως χρησιμοποιούν πιο παραδοσιακούς ευρετικούς αλγόριθμους σαν υποδιαδικασίες τους. Πολλές φορές μπορούν να επιτραπούν, σε κάποιο μεθευρετικό αλγόριθμο, βήματα που οδηγούν σε μη εφικτή ενδιάμεση λύση σε κάποιο βήμα του αλγορίθμου. Ο λόγος που επιτρέπεται αυτό είναι για να αποφευχθεί κάποιο τοπικό ελάχιστο και η ολική λύση που θα εξαχθεί από τον αλγόριθμο να είναι καλύτερη. Ένα επιπλέον χαρακτηριστικό αυτών των αλγορίθμων είναι ότι προσομοιάζουν μια διαδικασία που συνήθως έχει εφαρμογή στη φύση [1].

Τα στοιχεία που χρησιμοποιούν αυτοί οι αλγόριθμοι και μεταφορικά τα παρατηρούμε και στη φύση είναι τα εξής:

- Χρησιμοποιούν ένα αριθμό από επαναληπτικές δοκιμές
- Περιλαμβάνουν ένα ή περισσότερους πράκτορες (νευρώνες, μόρια, χρωμοσώματα, μυρμήγκια)
- Λειτουργούν (στην περίπτωση των πολύ – πρακτόρων) βάση ενός μηχανισμού συνεργασίας και ανταγωνισμού
- Περιλαμβάνουν διαδικασίες αυτό – τροποποιήσεων των ευρετικών παραμέτρων ή ακόμα και της αναπαράστασης του προβλήματος.

Τα χαρακτηριστικά των μεθευρετικών είναι τα εξής:

- Μοντελοποιούν ένα φαινόμενο που υπάρχει στη φύση
- Μπορούν να μεταφερθούν εύκολα σε παράλληλη μορφή
- Είναι προσαρμοστικοί αλγόριθμοι.

3.2. Ευρετικοί αλγόριθμοι που εφαρμόζονται στη διπλωματική εργασία

Στα υποκεφάλαια 3.2.1 και 3.2.2 αναφέρονται οι μεθοδολογίες των αλγορίθμων **πλησιέστερου γείτονα** και **2-OPT**. Ενώ τα χαρακτηριστικά του αλγορίθμου **βελτιστοποίησης αποικίας μυρμηγκιών** παρουσιάζονται στο 3.3.

3.2.1. Αλγόριθμος πλησιέστερου γείτονα

Σε αυτήν την ευρετική διαδικασία, ο πωλητής ξεκινά από κάποια πόλη και μετά επισκέπτεται την πόλη που είναι πλησιέστερη στην αρχική. Από εκεί επισκέπτεται την πλησιέστερη πόλη που δεν έχει ακόμη επισκεφτεί, μέχρι ότου όλες οι πόλεις συμπεριληφθούν στην διαδρομή του, και τότε επιστρέφει στην αρχική πόλη. Τα βασικά βήματα του αλγορίθμου είναι:

Βήμα 1: Ξεκίνα με οποιοδήποτε κόμβο σαν ξεκίνημα της διαδρομής

Βήμα 2: Βρες τον πλησιέστερο κόμβο που δεν συμπεριλαμβάνεται στη διαδρομή και πρόσθεσέ τον στο μονοπάτι.

Βήμα 3: Επανάλαβε το Βήμα 2 μέχρις ότου όλοι οι κόμβοι ανήκουν στην διαδρομή [1].

3.2.2. Αλγόριθμος 2-OPT

Η συγκεκριμένη μέθοδος αποτελείται γενικά από τη διαγραφή δυο ακμών και την επανασύνδεση δυο μονοπατιών με διαφορετικό τρόπο για να καθοριστεί μια καινούργια διαδρομή. Υπάρχει μόνο ένας τρόπος για να επανασυνδεθούν τα μονοπάτια.

Η διαδικασία της μεθόδου για μία συγκεκριμένη διαδρομή έχει ως εξής:

- Πρώτα, για κάθε κόμβο $i=1\dots,n$: Εξετάζει όλες τις πιθανές 2-opt κινήσεις που μπορεί να γίνουν από την i και την επόμενη της μέσα στην διαδρομή.
- Αν με αυτόν τον τρόπο είναι δυνατή η μείωση του κόστους της διαδρομής, τότε επιλέγει την καλύτερη 2-opt κίνηση και εφαρμόζει τις αλλαγές στην διαδρομή.
- Αυτή η διαδικασία επαναλαμβάνεται έως ότου φτάσει σε σημείο που είναι αδύνατο να πραγματοποιηθεί επιπλέον βελτίωση [1].

Έτσι έστω για παράδειγμα η αρχική λύση:

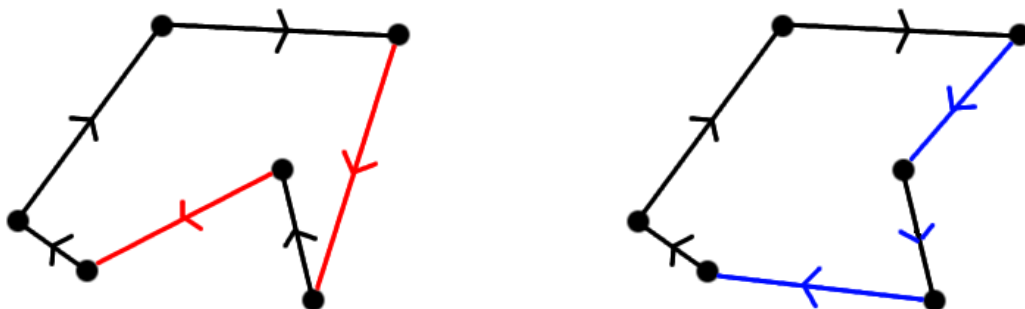
[... i , j , k , l , m , n , o , p , ...]

και οι διαδρομές προς κατάργηση είναι οι $j \rightarrow k$ και $n \rightarrow o$, οι οποίες ανήκουν στις ακριβότερες διαδρομές της συγκεκριμένης λύσης. Το νέο διάνυσμα που θα προκύψει, η νέα λύση δηλαδή, θα διαφέρει από την αρχική μόνο στα στοιχεία που βρίσκονται ανάμεσα στα στοιχεία j και o και θα είναι η εξής:

$$[\dots i, j, \underline{n}, \underline{m}, \underline{l}, k, o, p, \dots].$$

Η νέα λύση όμως δεν είναι σίγουρα καλύτερη της αρχικής, για αυτό το λόγο υπολογίζεται το νέο κόστος. Αν το κόστος της συγκεκριμένης λύσης είναι καλύτερο από το κόστος της αρχικής τότε γίνεται δεκτό, εάν όχι τότε επιλέγεται μία διαδρομή από τις ακριβότερες και επιλέγεται μία δεύτερη διαδρομή, εντελώς τυχαία, η οποία καταργείται. Μετά δημιουργείται η νέα λύση όπως περιγράφεται παραπάνω.

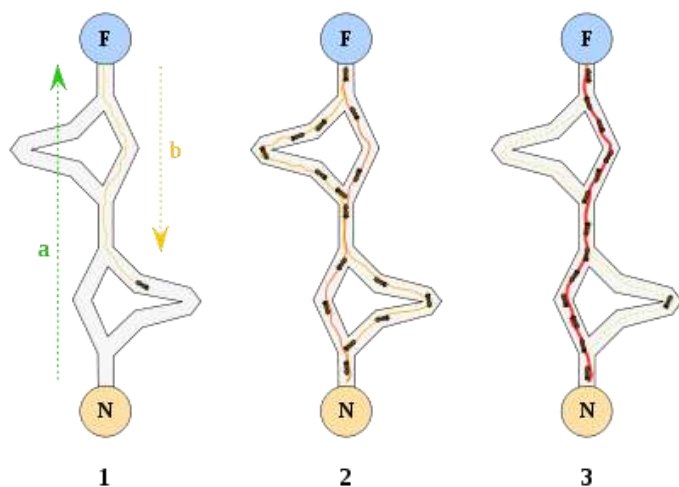
Στη συνέχεια γίνεται έλεγχος πάλι του κόστους της νέας εφικτής λύσης. Εάν το συγκεκριμένο κόστος είναι καλύτερο από αυτό της αρχικής τότε η νέα λύση γίνεται αποδεκτή, εάν όχι τότε παραμένει ως καλύτερη η προηγούμενη αρχική λύση. Στο επόμενο βήμα αναζητείται νέα λύση, θέτοντας σαν αρχική λύση, τη λύση που έγινε αποδεκτή στο προηγούμενο βήμα. Αυτή η διαδικασία επαναλαμβάνεται μέχρι ένα σημείο όπου η λύση δεν θα μπορεί να βελτιωθεί άλλο ή μέχρι να συμπληρωθεί ο μέγιστος αριθμός βημάτων της μεθόδου 2-opt. Τέλος επιστρέφεται σαν αποτέλεσμα μία λύση η οποία είναι καλύτερη από την αρχική, αν βέβαια έχει επέλθει βελτίωση. Αν δεν βελτιωθεί η αρχική λύση σε κανένα βήμα της διαδικασίας τότε επιστρέφεται σαν αποτέλεσμα η αρχική λύση.



Σχηματική απεικόνιση εφαρμογής 2-OPT

3.3. Αλγόριθμος βελτιστοποίησης αποικίας μυρμηγκιών (Ant Colony Optimization – ACO)

Η βελτιστοποίηση με βάση αποικίες μυρμηγκιών είναι ένα σύστημα που μιμείται την συμπεριφορά των πραγματικών μυρμηγκιών κατά τη διαδικασία εύρεσης της τροφής τους. Στην πραγματικότητα τα μυρμήγκια ξεκινούν τυχαία στο χώρο προς εύρεση τροφής και όταν βρουν τροφή την μεταφέρουν στη φωλιά τους αφήνοντας πίσω τους μία χημική ουσία, την φερομόνη. Με αυτόν τον τρόπο είναι πολύ πιθανό το επόμενο μυρμήγκι να ακολουθήσει την ίδια διαδρομή



προς την τροφή. Όσο πιο σύντομη είναι η διανυθείσα απόσταση και περισσότερη η τροφή που βρέθηκε, τόσο περισσότερη φερομόνη αφήνουν στο μονοπάτι. Έτσι η ποσότητα φερομόνης στο βέλτιστο μονοπάτι αυξάνεται συνεχώς αφού όλο και περισσότερα μυρμήγκια ακολουθούν την συγκεκριμένη διαδρομή. Αντίθετα, στα υπόλοιπα υποβέλτιστα μονοπάτια η φερομόνη ελαττώνεται καθώς δεν περνούν άλλα μυρμήγκια.

Στον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών το πρόβλημα μοντελοποιείται ως πρόβλημα εύρεσης μονοπατιού ελαχίστου κόστους σε ένα γράφημα. Κάθε μυρμήγκι αποτελεί μια ξεχωριστή λύση για το πρόβλημα. Καθένα ξεκινάει να κατασκευάζει μία λύση με βάση την εμπειρία-φερομόνη που έχει αποκτηθεί από τις προηγούμενες λύσεις-μυρμήγκια. Στο τέλος των επαναλήψεων σχεδόν όλα τα μυρμήγκια ακολουθούν την ίδια διαδρομή η οποία κρατείται ως βέλτιστη.

3.3.1. Αλγόριθμος βελτιστοποίησης αποικίας μυρμηγκιών στο πρόβλημα του πλανόδιου πωλητή – TSP (Travelling Salesman Problem)

Το βασικό χαρακτηριστικό των τεχνητών συστημάτων μυρμηγκιών είναι η ιδιότητα να έχουν μνήμη (data structures). Επίσης έχουν την δυνατότητα να «αισθάνονται» και να αντιδρούν ανάλογα με το περιβάλλον. Επιπρόσθετα, είναι συστήματα διακριτού χρόνου και αλγόριθμοι βελτιστοποίησης.

Πιο συγκεκριμένα στο πρόβλημα του πλανόδιου πωλητή η φερομόνη τ_{ij} συμβολίζει την επιθυμία ο πωλητής να επισκεφτεί την πόλη j αμέσως μετά την i . Κατασκευάζεται ένας πίνακας:

$$n_{ij} = \frac{1}{c_{ij}} \quad (3.3.1)$$

όπου c_{ij} είναι η απόσταση από τον κόμβο i στον j . Έτσι ο πίνακας n_{ij} ή αλλιώς πίνακας ευρετικής πληροφορίας δείχνει την αντιστρόφως ανάλογη σχέση μεταξύ της επιθυμίας του πωλητή να επισκεφθεί την πόλη j ενώ βρίσκεται στην i και του κόστους μετάβασης $i \rightarrow j$. Αρχικά κάθε μυρμήγκι κατασκευάζει μία διαδρομή-λύση ξεκινώντας από διαφορετικές πόλεις. Κάθε μυρμήγκι διαλέγει σε ποια πόλη θα πάει με βάση μία πιθανότητα, η οποία είναι συνάρτηση της απόστασής της από την προηγούμενη πόλη και της υπάρχουσας στο μονοπάτι αυτό φερομόνης. Η μετάβαση από έναν κόμβο-πόλη που έχει ήδη εξυπηρετηθεί απαγορεύεται. Όταν η διαδρομή ενός μυρμηγκιού τελειώσει τότε τοποθετείται στις διανυθείσες διαδρομές φερομόνη αντιστρόφως ανάλογη του κόστους της λύσης-μονοπατιού. Ένας αρχικός τρόπος για τον υπολογισμό της αρχικής φερομόνης στα τόξα είναι:

$$\tau_{ij} = \frac{m}{TC} \quad (3.3.2)$$

όπου m είναι το πλήθος των μυρμηγκιών και TC είναι το κόστος ενός αρχικού κύκλου που στην παρούσα διπλωματική έχει κατασκευαστεί με βάση τον αλγόριθμο του πλησιέστερου γείτονα. Το ποσό της αρχικής φερομόνης έχει μεγάλη σημασία διότι αν η αρχική φερομόνη είναι πολύ μικρή, τότε υπάρχει η περίπτωση μία σχετικά καλή αρχική λύση ενός μυρμηγκιού να κυριαρχήσει και να οδηγήσει τον αλγόριθμο σε τοπικό βέλτιστο. Αντιθέτως εάν η ποσότητα της αρχικής φερομόνης είναι μεγάλη θα πρέπει να περάσει ένας μεγάλος αριθμός επαναλήψεων έως ότου η προσθήκη φερομόνης από τα μυρμήγκια να επηρεάσει τη λύση.

Όπως προαναφέρθηκε υπάρχει ένας πιθανοτικός κανόνας που ορίζει την πιθανότητα μετάβασης p_{ij} από έναν κόμβο i στον j :

$$p_{ij} = \frac{[\tau_{ij}]^\alpha [n_{ij}]^\beta}{\sum_{l=1}^M [\tau_{il}]^\alpha [n_{il}]^\beta} \quad (3.3.3)$$

όπου M είναι το σύνολο των κόμβων που δεν έχουν εξυπηρετηθεί και α, β δύο παράμετροι που καθορίζουν αν επηρεάζει περισσότερο την επιλογή η φερομένη που έχει το τρέχον τόξο ή η ευρετική πληροφορία n_{ij} αυτού. Για παράδειγμα εάν $\alpha=0$ είναι πιθανότερο να επιλεγούν οι κοντινότερες πόλεις, ενώ εάν $\beta=0$ τότε μόνο η φερομένη χρησιμοποιείται, κάτι που δεν οδηγεί σε καλά αποτελέσματα.

Ακολουθώντας, ενώ όλα τα μυρμήγκια έχουν κατασκευάσει τους αρχικούς κύκλους γίνεται ενημέρωση και η αναβάθμιση του πίνακα με τις φερομένες τ_{ij} . Ένας τρόπος είναι αρχικά να μειωθεί η φερομένη κατά $(1-\rho)\tau_{ij}$, όπου $0 < \rho < 1$ είναι ένας συντελεστής εξάτμισης, έτσι ώστε κακές λύσεις να απορρίπτονται κατά την εκτέλεση του αλγορίθμου. Στη συνέχεια η φερομένη των τόξων που χρησιμοποιήθηκαν αυξάνεται κατά:

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3.3.4)$$

όπου $\Delta\tau_{ij}^k$ είναι η ποσότητα φερομένης που αφήνει κάθε μυρμήγκι στα τόξα που πέρασε και δίνεται από τον τύπο:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{1}{C^k}, & \text{εάν το τόξο έχει επιλεγεί από το } k \text{ μυρμήγκι} \\ 0, & \text{αλλιώς} \end{cases} \quad (3.3.5)$$

όπου C^k το κόστος του κύκλου που έχει δημιουργήσει το μυρμήγκι k . Έτσι η ποσότητα φερομένης που αφήνει κάθε μυρμήγκι είναι αντιστρόφως ανάλογη του κόστους. Δηλαδή, όσο καλύτερος είναι ο κύκλος που έχει δημιουργήσει τόσο περισσότερη φερομένη προστίθεται στη διανυθείσα διαδρομή. Η στρατηγική που ακολουθείται στην παρούσα διπλωματική εργασία όσον αφορά την διαδικασία ενημέρωσης της φερομένης είναι να αφήνουν φερομένη όλα τα μυρμήγκια και το βέλτιστο μυρμήγκι μία επιπλέον ποσότητα[1].

Κεφάλαιο 4 – Εφαρμογή ACO στο OVRP

4.1. Παρουσίαση του προβλήματος προς βελτιστοποίηση

Στα πλαίσια αυτής της διπλωματικής επιλύεται το πρόβλημα δρομολόγησης οχημάτων χωρίς επιστροφή αυτών στην αποθήκη. Επιπλέον, τα οχήματα είναι πεπερασμένης χωρητικότητας. Άρα το προς επίλυση πρόβλημα είναι το «Πρόβλημα Δρομολόγησης Οχημάτων Χωρίς Επιστροφή των Οχημάτων στην Αποθήκη» ή όπως αναφέρεται στην παγκόσμια βιβλιογραφία Open Vehicle Routing Problem.

Ο αλγόριθμος που αναπτύσσεται και επιλύει το προαναφερθέν πρόβλημα εφαρμόζεται σε μία σειρά παραδειγμάτων που έχουν ήδη επιλυθεί στα [6], [7], [8]. Τα δεδομένα των παραδειγμάτων είναι:

- ο αριθμός των κόμβων (n) χωρίς την αποθήκη,
- η χωρητικότητα (C) των οχημάτων,
- ο μέγιστος χρόνος (L) που δύναται ένα όχημα να παραμείνει στη διαδρομή και
- ο χρόνος εξυπηρέτησης (e) κάθε πελάτη.

Πιο αναλυτικά για τα 14 παραδείγματα:

Παραδείγματα	n	C	L	e
C1	50	160		
C2	75	140		
C3	100	200		
C4	150	200		
C5	199	200		
C6	50	160	180	10
C7	75	140	144	10
C8	100	200	207	10
C9	150	200	180	10
C10	199	200	180	10
C11	120	200		
C12	100	200		
C13	120	200	648	50
C14	100	200	936	90

Πίνακας 4.1

Λεπτομερής παρουσίαση του κάθε παραδείγματος διατίθεται στο Παράρτημα 1.

Στην στήλη L , όπου δεν υπάρχει αριθμός θεωρούμε ότι ο χρόνος που μπορεί να παραμείνει στη διαδρομή ένα όχημα είναι άπειρος. Συνεπώς, σε αυτά τα παραδείγματα δεν έχει λόγο ύπαρξης

και ο χρόνος εξυπηρέτησης. Με βάση τα δοθέντα στοιχεία μπορεί εύκολα να υπολογισθεί ο ελάχιστος απαιτούμενος αριθμός οχημάτων (K_{min}) σε κάθε παράδειγμα. Αυτός θα δίνεται ως το πηλίκο του αθροίσματος της ζήτησης κάθε πελάτη προς τη χωρητικότητα του οχήματος ή

$$K_{min} = \lceil \frac{\sum_{i=1}^n d_i}{C} \rceil$$

όπου d_i η ζήτηση του πελάτη i .

Για παράδειγμα, στην C2 περίπτωση είναι:

$$\sum_{i=1}^n d_i = 1327,$$

$$C=140,$$

έτσι:

$$K_{min} = \lceil 9,478571 \rceil = 10 \text{ οχήματα}$$

Συνεπώς ο πίνακας 4.1 γίνεται:

Παραδείγματα	n	C	L	e	K_{min}
C1	50	160			5
C2	75	140			10
C3	100	200			8
C4	150	200			12
C5	199	200			16
C6	50	160	180	10	5
C7	75	140	144	10	10
C8	100	200	207	10	8
C9	150	200	180	10	12
C10	199	200	180	10	16
C11	120	200			7
C12	100	200			10
C13	120	200	648	10	7
C14	100	200	936	10	10

Πίνακας 4.2

4.2. Βέλτιστες λύσεις των παραδειγμάτων σε παγκόσμια κλίμακα

Τα προαναφερθέντα παραδείγματα έχουν επιλυθεί και οι λύσεις τους έχουν συγκριθεί στα [6], [7], [8], [9], [10] και [11]. Έτσι τα μέχρι στιγμής βέλτιστα αποτελέσματα είναι τα κάτωθι:

Παραδείγματα Προβλήματος	Κόστος	Αριθμός Οχημάτων
C1	416.06	5
C2	567.14	10
C3	639.74	8
C4	733.13	12
C5	894.11	16
C6	412.96	6
C7	584.15	10
C8	644.63	9
C9	764.56	13
C10	888.46	17
C11	682.12	7
C12	534.24	10
C13	910.26	11
C14	591.87	11

Πίνακας 4.3

Οι παραπάνω λύσεις έχουν προκύψει από την εφαρμογή της υβριδικής στρατηγικής, HES-hybrid evolution strategy for the open vehicle routing problem που προτείνεται στο [6].

4.3. Αλγοριθμική προσέγγιση προβλήματος

Σε αυτό το κεφάλαιο παρουσιάζεται ένας ψευδοκώδικας του αλγορίθμου που εφαρμόστηκε για την επίλυση του δοθέντος προβλήματος. Το θεωρητικό υπόβαθρο για την κατανόηση της μεθοδολογίας αναφέρθηκε στο υποκεφάλαιο 3.2.

Αρχικοποίηση

Δημιουργία αρχικού πληθυσμού μυρμηγκιών

Υπολογισμός ευρετικής συνάρτησης n_{ij} για κάθε δυνατό μονοπάτι

Εφαρμογή **αλγορίθμου πλησιέστερου γείτονα**, τόσες φορές όσοι και οι κόμβοι, για τον υπολογισμό μίας αρχικής, υποβέλτιστης λύσης

Υπολογισμός αρχικής φερομόνης τ_{ij} για κάθε δυνατό μονοπάτι

Κύρια φάση

While δεν έχει φτάσει ο μέγιστος αριθμός επαναλήψεων

Κάθε μυρμήγκι ξεκινάει την πορεία του από διαφορετικό κόμβο

For κάθε μυρμήγκι

While ικανοποιούνται οι περιορισμοί

Επιλογή επόμενου κόμβου που θα επισκεφτεί το μυρμήγκι βάση του πίνακα πιθανοτήτων που κατασκευάζεται σε κάθε κόμβο του συστήματος, συνάρτηση της φερομόνης τ_{ij} και της ευρετικής πληροφορίας n_{ij}

If υπάρχουν ανεξυπηρέτητοι πελάτες

Στείλε νέο όχημα για την εξυπηρέτηση των υπολοίπων πελατών

Endif

Endwhile

Endfor

Εύρεση νέων λύσεων

For τα καλύτερα μυρμήγκια μυρμήγκια

Do until βελτιωθεί η λύση ή φτάσει το μέγιστο αριθμό επαναλήψεων

Εφαρμογή αλγορίθμου **2-OPT**

Enddo

Endfor

Update φερομόνης βάση των νέων λύσεων

Μείωση της φερομόνης σε όλα τα τόξα

Αύξηση της φερομόνης στα τόξα που προσπελάστηκαν από το βέλτιστο μυρμήγκι

Endwhile

Return βέλτιστη λύση, βέλτιστο μονοπάτι

Οι περιορισμοί του συγκεκριμένου προβλήματος είναι οι ακόλουθοι:

- Η ζήτηση του επόμενου πελάτη να είναι μικρότερη από την υπολειπόμενη χωρητικότητα του οχήματος.
- Το άθροισμα του χρόνου μετάβασης από τον τρέχοντα πελάτη έως τον επόμενο και του χρόνου εξυπηρέτησης αυτού να είναι μικρότερο από τον υπολειπόμενο χρόνο του οχήματος στη διαδρομή.

Στο επόμενο κεφάλαιο παρουσιάζονται τα αποτελέσματα της εφαρμογής του αλγορίθμου στα 14 παραδείγματα του 4.1.

Κεφάλαιο 5 – Αποτελέσματα ACO στο OVRP

5.1. Παράδειγμα C1

Σε αυτό το παράδειγμα το πρόβλημα αποτελείται από $n=50$ κόμβους προς εξυπηρέτηση. Υπάρχει περιορισμός στην χωρητικότητα του οχήματος, $C=160$. Περιορισμός ως προς το μέγιστο χρόνο που δύναται ένα όχημα να παραμείνει στη διαδρομή και ως προς το χρόνο εξυπηρέτησης κάθε πελάτη δεν υπάρχει σε αυτό το παράδειγμα όπως και στα C2, C3, C4, C5, C11 και C12.

Εφαρμόζοντας αρχικά τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor) προκύπτει:

- $MinCost=492.5$

Όπου με $MinCost$ θα συμβολίζεται το ελάχιστο κόστος για την κάλυψη της ζήτησης όλων των πελατών.

5.1.1. Αποτελέσματα εφαρμογής ACO και 2-OPT στο παράδειγμα 1

Εφαρμόζοντας τον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών προκύπτει:

- $MinCost=451,2$
- $K=5$

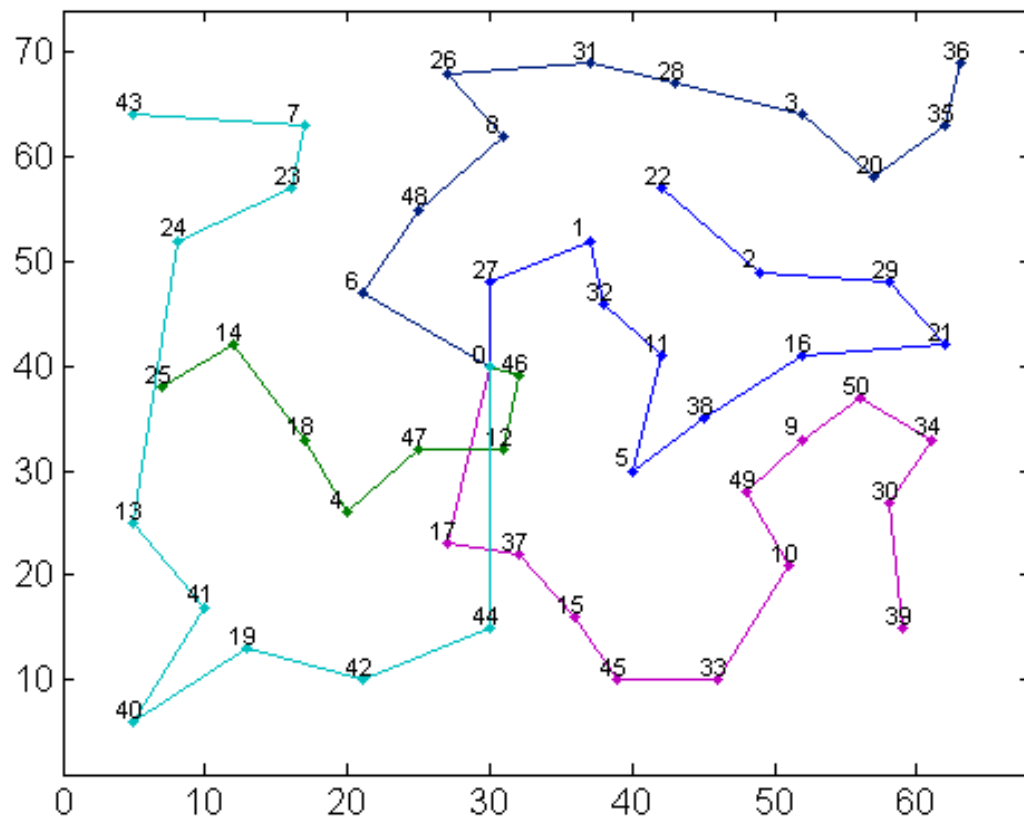
Το μονοπάτι που οδήγησε σε αυτό το κόστος είναι το:

[0 17 37 15 45 33 10 49 9 50 34 30 39 0 27 1 32 11 5 38 16 21 29 2 22 0 6 48 8 26 31 28 3 20 35 36 0 46 12 47 4 18 14 25 0 44 42 19 40 41 13 24 23 7 43] ,

ή ποιο συγκεκριμένα, για κάθε όχημα:

0 17 37 15 45 33 10 49 9 50 34 30 39
0 27 1 32 11 5 38 16 21 29 2 22
0 6 48 8 26 31 28 3 20 35 36
0 46 12 47 4 18 14 25
0 44 42 19 40 41 13 24 23 7 43

5.1.2. Γραφική απεικόνιση λύσης παραδείγματος 1



5.2. Παράδειγμα C2

Όμοια με το C2, περιορισμός ως προς το μέγιστο χρόνο που δύναται ένα όχημα να παραμείνει στη διαδρομή και ως προς το χρόνο εξυπηρέτησης κάθε πελάτη δεν υπάρχει ούτε σε αυτό το παράδειγμα. Για $n=75$ και $C=140$, εφαρμόζοντας αρχικά τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor) προκύπτει:

- MinCost=810

5.2.1. Αποτελέσματα εφαρμογής ACO και 2-OPT στο παράδειγμα 2

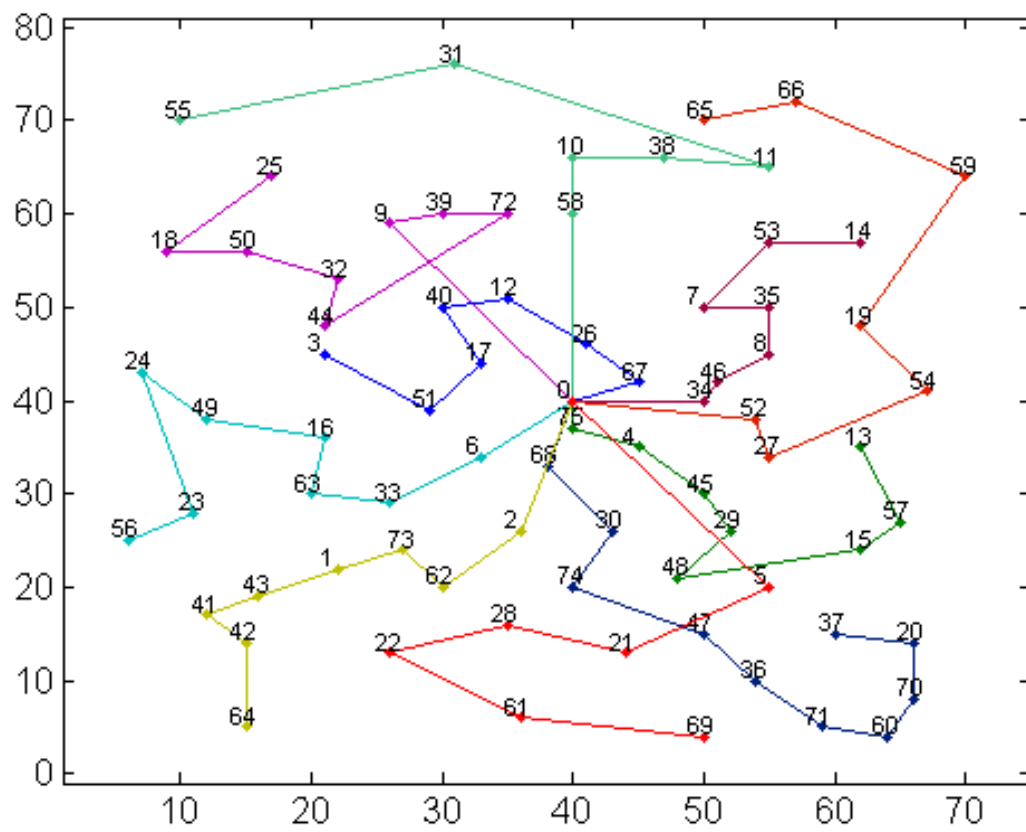
Εφαρμόζοντας τον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών προκύπτει:

- MinCost=694,21
- $K=10$

Η διαδρομή καθενός εκ των 10 οχημάτων:

0	58	10	38	11	31	55				
0	9	39	72	44	32	50	18	25		
0	6	33	63	16	49	24	23	56		
0	2	62	73	1	43	41	42	64		
0	28	30	74	47	36	71	60	70	20	37
0	75	4	45	29	48	15	57	13		
0	52	27	54	19	59	66	65			
0	34	46	8	35	7	53	14			
0	67	26	12	40	17	51	3			
0	5	21	28	22	61	69				

5.2.2. Γραφική απεικόνιση λύσης παραδείγματος 2



5.3. Παράδειγμα C3

Στο παράδειγμα C3, $n=100$ και $C=200$. Εφαρμόζοντας αρχικά τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor) προκύπτει:

- MinCost=871

5.3.1. Αποτελέσματα εφαρμογής ACO και 2-OPT στο παράδειγμα 3

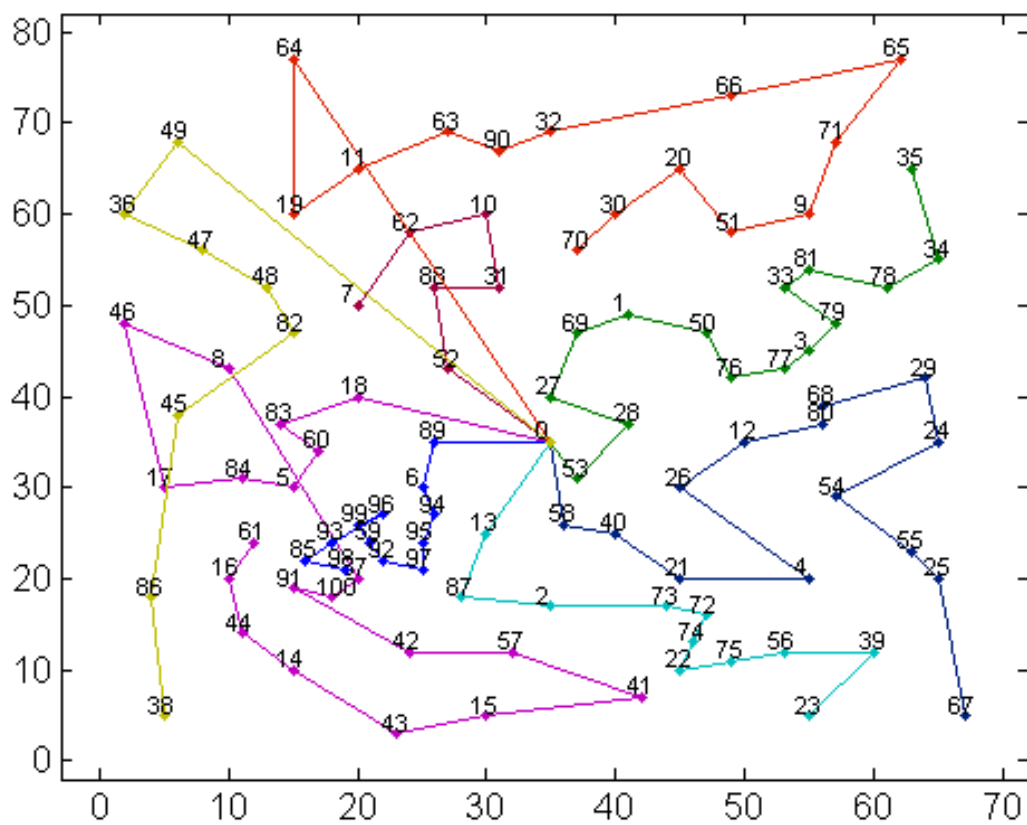
Εφαρμόζοντας τον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών προκύπτει:

- MinCost=811,46
- $K=8$

Η διαδρομή καθενός εκ των 8 οχημάτων:

0	18	83	60	5	84	17	46	8	37	100	91	42	57	41	15	43	14	44	16	61
0	53	28	27	69	1	50	76	77	3	79	33	81	78	34	35					
0	89	6	94	95	97	92	59	99	96	93	85	98								
0	58	40	21	4	26	12	80	68	29	24	54	55	25	67						
0	13	87	2	73	72	74	22	75	56	39	23									
0	52	88	31	10	62	7														
0	64	19	11	63	90	32	66	65	71	9	51	20	30	70						
0	49	36	47	48	82	45	86	38												

5.3.2. Γραφική απεικόνιση λύσης παραδείγματος 3



5.4. Παράδειγμα C4

Στο παράδειγμα C4, $n=150$ και $C=200$. Εφαρμόζοντας αρχικά τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor) προκύπτει:

- $\text{MinCost}=1018.5$

5.4.1. Αποτελέσματα εφαρμογής ACO και 2-OPT στο παράδειγμα 4

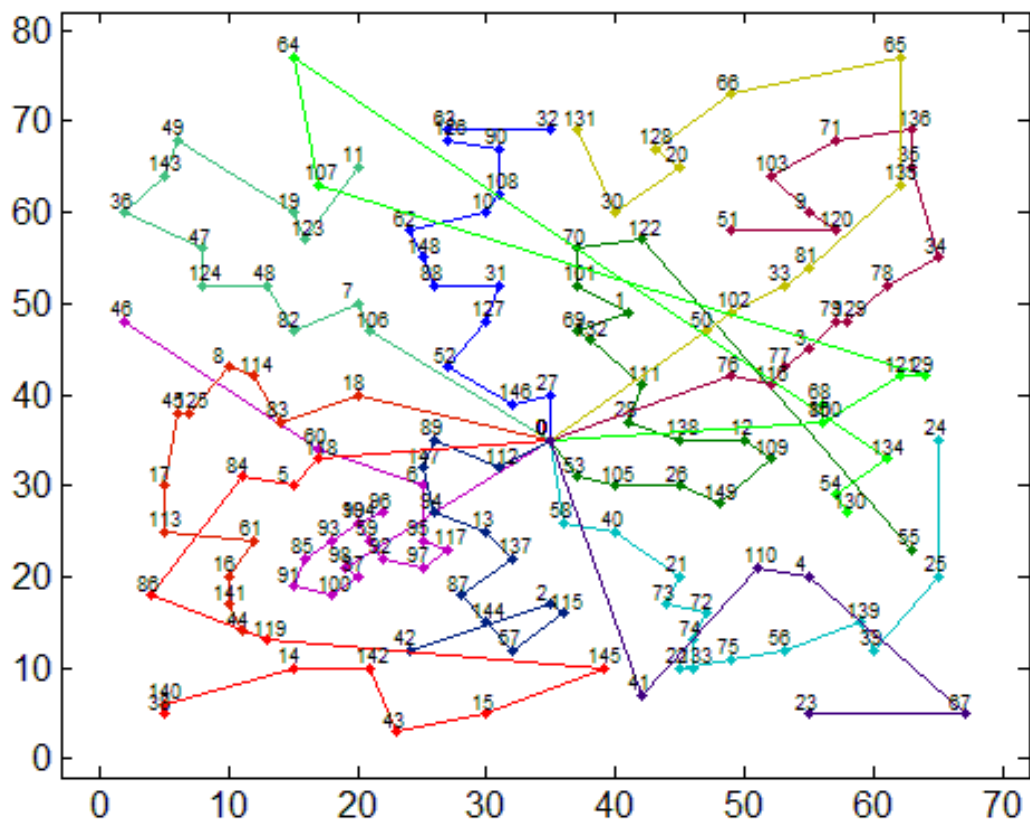
Εφαρμόζοντας τον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών προκύπτει:

- $\text{MinCost}=898,46$
- $K=12$

Η διαδρομή καθενός εκ των 12 οχημάτων:

0	98	37	100	91	85	93	99	96	104	59	92	97	117	95	6	60	46
0	53	105	26	149	109	12	138	28	111	132	69	1	101	70	122	55	
0	27	146	52	127	31	88	148	62	10	108	90	126	63	32			
0	112	89	147	94	13	137	87	144	57	115	2	42					
0	58	40	21	73	72	74	133	22	75	56	139	39	25	24			
0	76	116	77	3	79	129	78	34	35	136	71	103	9	120	51		
0	18	83	114	8	125	45	17	113	61	16	141	44					
0	50	102	33	81	135	65	66	128	20	30	131						
0	106	7	82	48	124	47	36	143	49	19	123	11					
0	118	5	84	86	119	145	15	43	142	14	140	38					
0	80	68	150	121	29	107	64	134	54	130							
0	41	110	4	67	23												

5.4.2. Γραφική απεικόνιση λύσης παραδείγματος 4



5.5. Παράδειγμα C5

Στο παράδειγμα C5, $n=199$ και $C=200$. Εφαρμόζοντας αρχικά τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor) προκύπτει:

- $\text{MinCost}=1247$

5.5.1. Αποτελέσματα εφαρμογής ACO και 2-OPT στο παράδειγμα 5

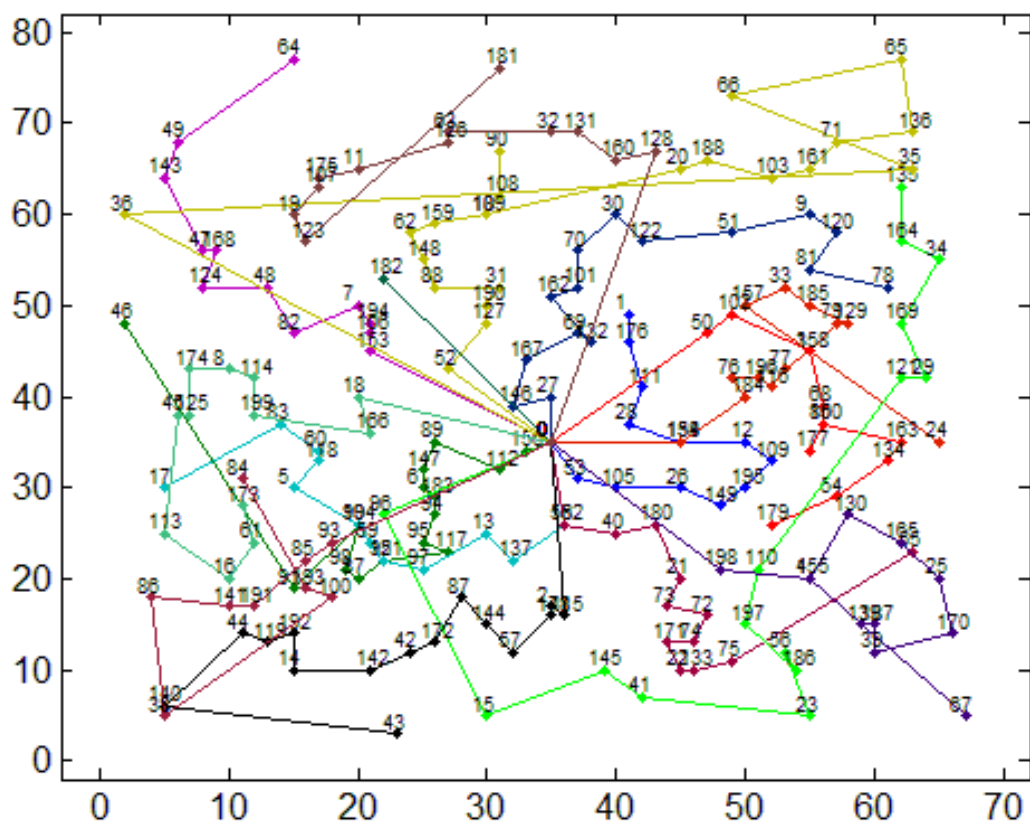
Εφαρμόζοντας τον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών προκύπτει:

- $\text{MinCost}=1106,18$
- $K=17$

Η διαδρομή καθενός εκ των 17 οχημάτων:

0	153	106	194	7	82	48	124	168	47	143	49	64			
0	156	112	89	147	6	183	94	95	117	92	37	98	104	91	46
0	53	105	26	149	195	109	12	138	28	111	176	1			
0	27	146	167	69	132	162	101	70	30	122	51	9	120	81	78
0	58	137	13	97	151	59	99	5	118	60	83	17			
0	152	40	180	21	73	72	74	171	22	133	75	55			
0	154	184	76	196	116	77	3	79	129	185	33	157	24		
0	52	127	190	31	88	148	62	159	189	108	90				
0	18	166	199	114	8	174	125	45	113	16	61	173			
0	50	102	158	68	80	177	150	163	134	54	179				
0	96	15	145	41	23	56	186	197	110	121	29	169	34	164	135
0	198	4	139	187	39	170	25	165	130	155	67				
0	115	2	178	57	144	87	172	42	142	14	192	119	44	140	43
0	93	85	191	141	86	38	100	193	84						
0	182														
0	36	35	66	65	136	71	161	103	188	20	10				
0	128	160	131	32	63	126	11	175	107	19	123	181			

5.5.2. Γραφική απεικόνιση λύσης παραδείγματος 5



5.6. Παράδειγμα C6

Σε αυτό το παράδειγμα το πρόβλημα αποτελείται από $n=50$ κόμβους προς εξυπηρέτηση. Υπάρχει περιορισμός στην χωρητικότητα του οχήματος, $C=160$. Επίσης υφίσταται και ο δεύτερος περιορισμός που αναφέρθηκε στο 4.3., ως προς το μέγιστο χρόνο που δύναται ένα όχημα να παραμείνει στη διαδρομή $L=180$ και ως προς το χρόνο εξυπηρέτησης κάθε πελάτη $e=10$ όπως και στα C7, C8, C9, C10, C13 και C14. Εφαρμόζοντας αρχικά τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor) προκύπτει:

- $\text{MinCost}=467$

5.6.1. Αποτελέσματα εφαρμογής ACO και 2-OPT στο παράδειγμα 6

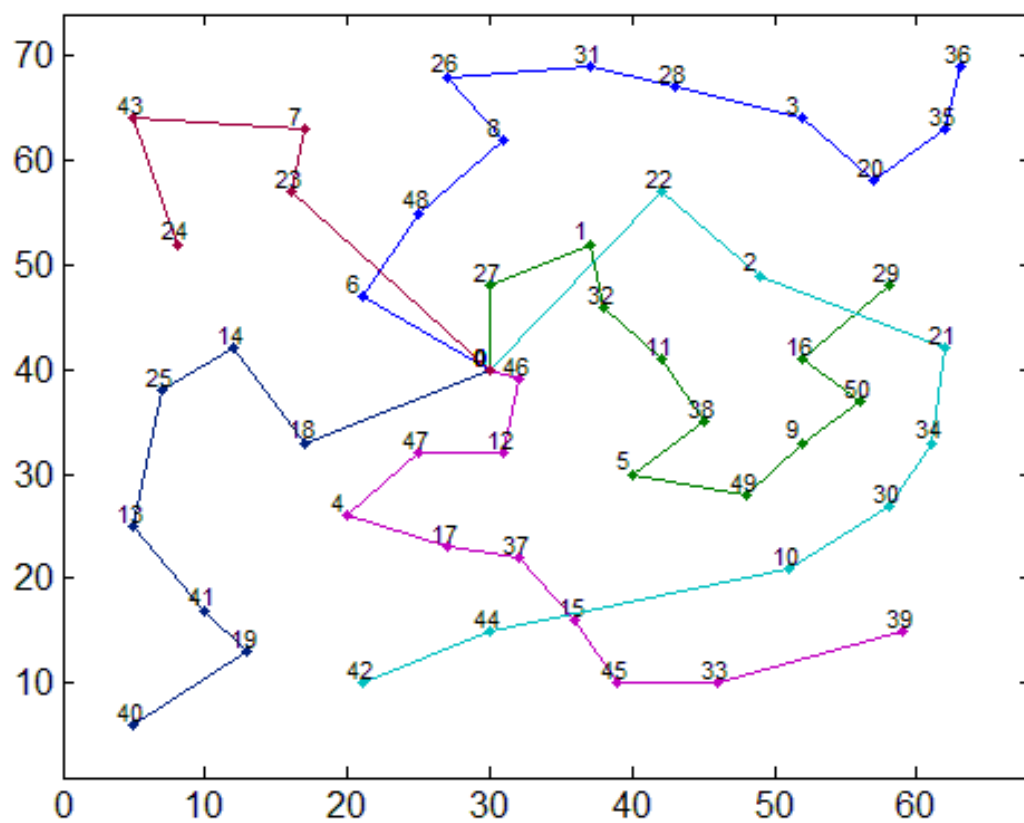
Εφαρμόζοντας τον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών προκύπτει:

- $\text{MinCost}=457,31$
- $K=6$

Η διαδρομή καθενός εκ των 6 οχημάτων:

0	46	12	47	4	17	37	15	45	33	39	
0	27	1	32	11	38	5	49	9	50	16	29
0	6	48	8	26	31	28	3	20	35	36	
0	18	14	25	13	41	19	40				
0	22	2	21	34	30	10	44	42			
0	23	7	43	24							

5.6.2. Γραφική απεικόνιση λύσης παραδείγματος 6



5.7. Παράδειγμα C7

Σε αυτό το παράδειγμα το πρόβλημα αποτελείται από $n=75$ κόμβους προς εξυπηρέτηση. Υπάρχει περιορισμός στην χωρητικότητα του οχήματος, $C=140$. Επίσης για τον δεύτερο περιορισμό, ο μέγιστος χρόνος που δύναται ένα όχημα να παραμείνει στη διαδρομή είναι $L=144$ και ο χρόνος εξυπηρέτησης κάθε πελάτη είναι $e=10$. Εφαρμόζοντας αρχικά τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor) προκύπτει:

- $\text{MinCost}=717$

5.7.1. Αποτελέσματα εφαρμογής ACO και 2-OPT στο παράδειγμα 7

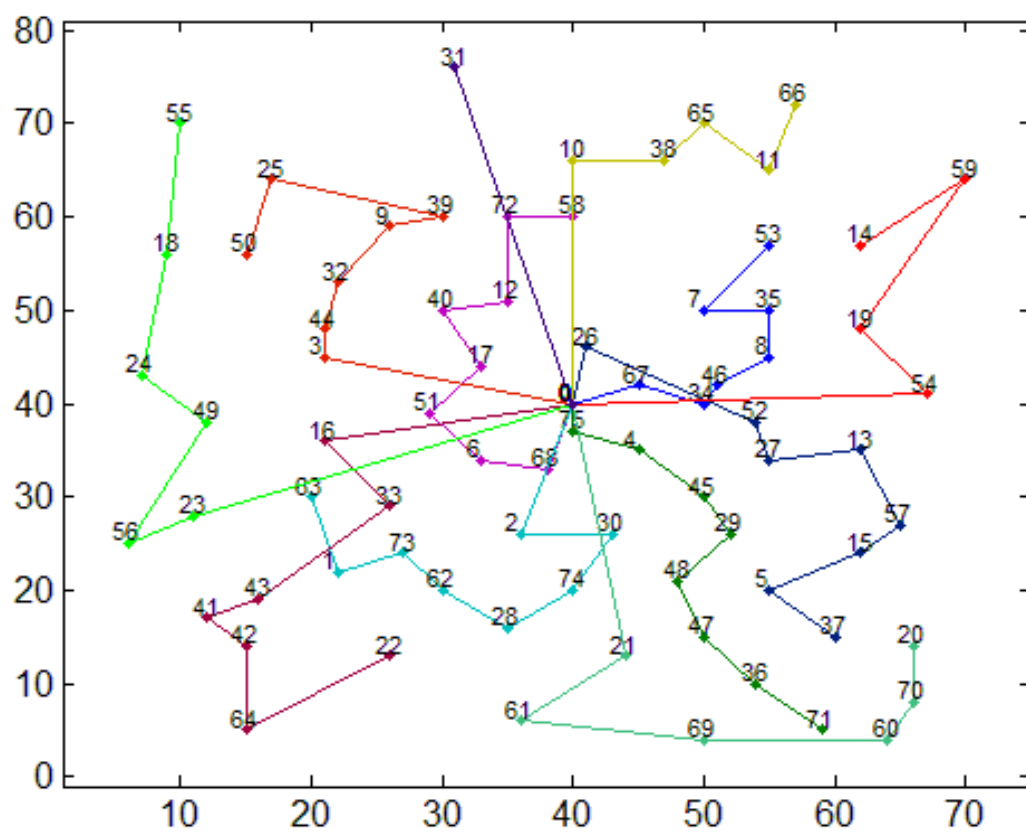
Εφαρμόζοντας τον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών προκύπτει:

- $\text{MinCost}=704,23$
- $K=12$

Η διαδρομή καθενός εκ των 12 οχημάτων:

0	68	6	51	17	40	12	72	58
0	75	4	45	29	48	47	36	71
0	67	34	46	8	35	7	53	
0	26	52	27	13	57	15	5	37
0	2	30	74	28	62	73	1	63
0	16	33	43	41	42	64	22	
0	10	38	65	11	66			
0	3	44	32	9	39	25	50	
0	21	61	69	60	70	20		
0	54	19	59	14				
0	23	56	49	24	18	55		
0	31							

5.7.2. Γραφική απεικόνιση λύσης παραδείγματος 7



5.8. Παράδειγμα C8

Στο παράδειγμα C8, $n=100$, $C=200$, $L=207$ και $e=10$. Εφαρμόζοντας αρχικά τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor) προκύπτει:

- $\text{MinCost}=765$

5.8.1. Αποτελέσματα εφαρμογής ACO και 2-OPT στο παράδειγμα 8

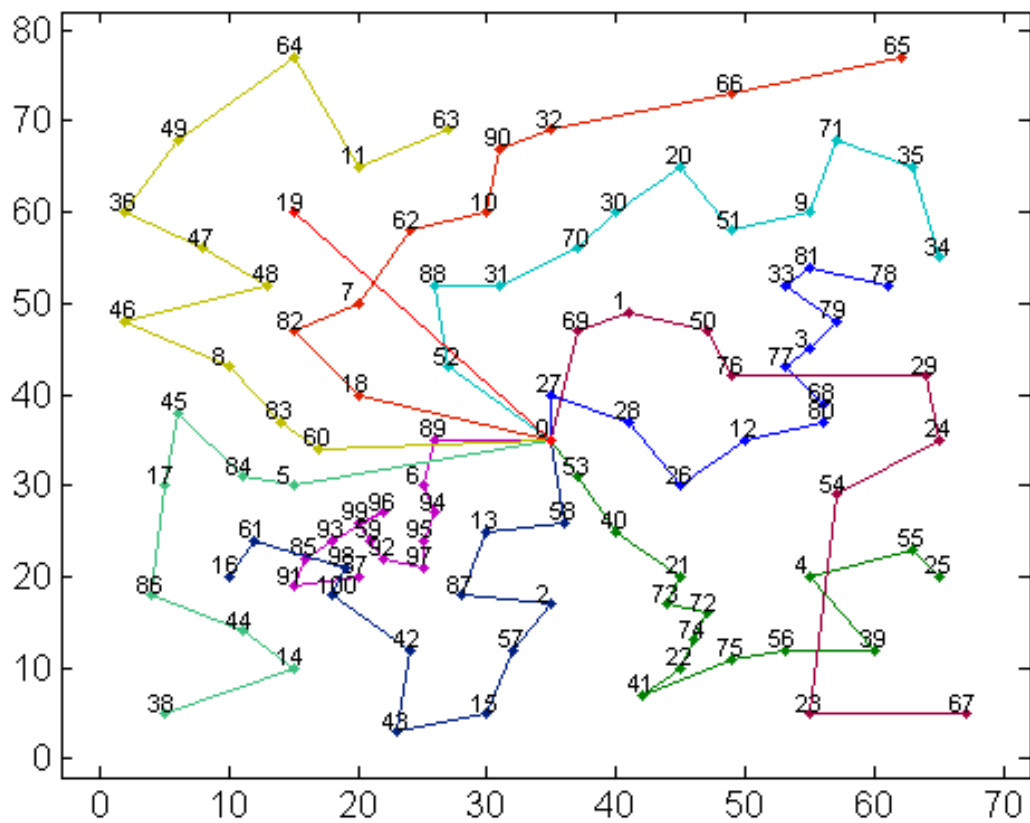
Εφαρμόζοντας τον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών προκύπτει:

- $\text{MinCost}=752,62$
- $K=10$

Η διαδρομή καθενός εκ των 10 οχημάτων:

0	89	6	94	95	97	92	59	99	96	93	85	91	37
0	27	28	26	12	80	68	77	3	79	33	81	78	
0	53	40	21	73	72	74	22	41	75	56	39	4	55
0	58	13	87	2	57	15	43	42	100	98	61	16	
0	52	88	31	70	30	20	51	9	71	35	34		
0	69	1	50	76	29	24	54	23	67				
0	18	82	7	62	10	90	32	66	65				
0	60	83	8	46	48	47	36	49	64	11	63		
0	5	84	45	17	86	44	14	38					
0	19												

5.8.2. Γραφική απεικόνιση λύσης παραδείγματος 8



5.9. Παράδειγμα C9

Στο παράδειγμα C9, $n=150$, $C=200$, $L=180$ και $e=10$. Εφαρμόζοντας αρχικά τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor) προκύπτει:

- $\text{MinCost}=893$

5.9.1. Αποτελέσματα εφαρμογής ACO και 2-OPT στο παράδειγμα 9

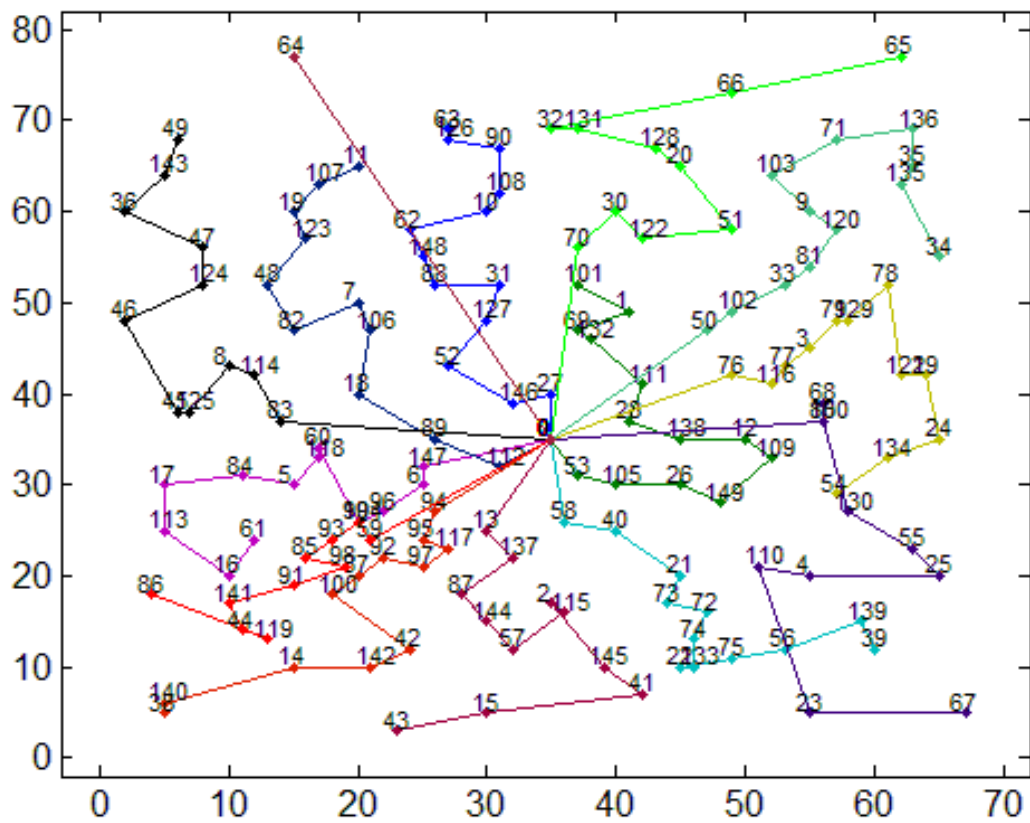
Εφαρμόζοντας τον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών προκύπτει:

- $\text{MinCost}=878,80$
- $K=14$

Η διαδρομή καθενός εκ των 14 οχημάτων:

0	6	96	104	60	118	5	84	17	113	16	61		
0	53	105	26	149	109	12	138	28	111	132	69	1	101
0	27	146	52	127	31	88	148	62	10	108	90	126	63
0	112	89	18	106	7	82	48	123	19	107	11		
0	58	40	21	73	72	74	133	22	75	56	139	39	
0	13	137	87	144	57	115	2	145	41	15	43		
0	94	95	117	97	92	37	100	42	142	14	140	38	
0	76	116	77	3	79	129	78	121	29	24	134	54	
0	50	102	33	81	120	9	103	71	136	35	135	34	
0	59	99	93	85	98	91	141	44	119	86			
0	70	30	122	51	20	128	131	32	66	65			
0	80	68	150	130	55	25	4	110	23	67			
0	83	114	8	125	45	46	124	47	36	143	49		
0	64												

5.9.2. Γραφική απεικόνιση λύσης παραδείγματος 9



5.10. Παράδειγμα C10

Στο παράδειγμα C10, $n=199$, $C=200$, $L=180$ και $e=10$. Εφαρμόζοντας αρχικά τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor) προκύπτει:

- $\text{MinCost}=1060$

5.10.1. Αποτελέσματα εφαρμογής ACO και 2-OPT στο παράδειγμα 10

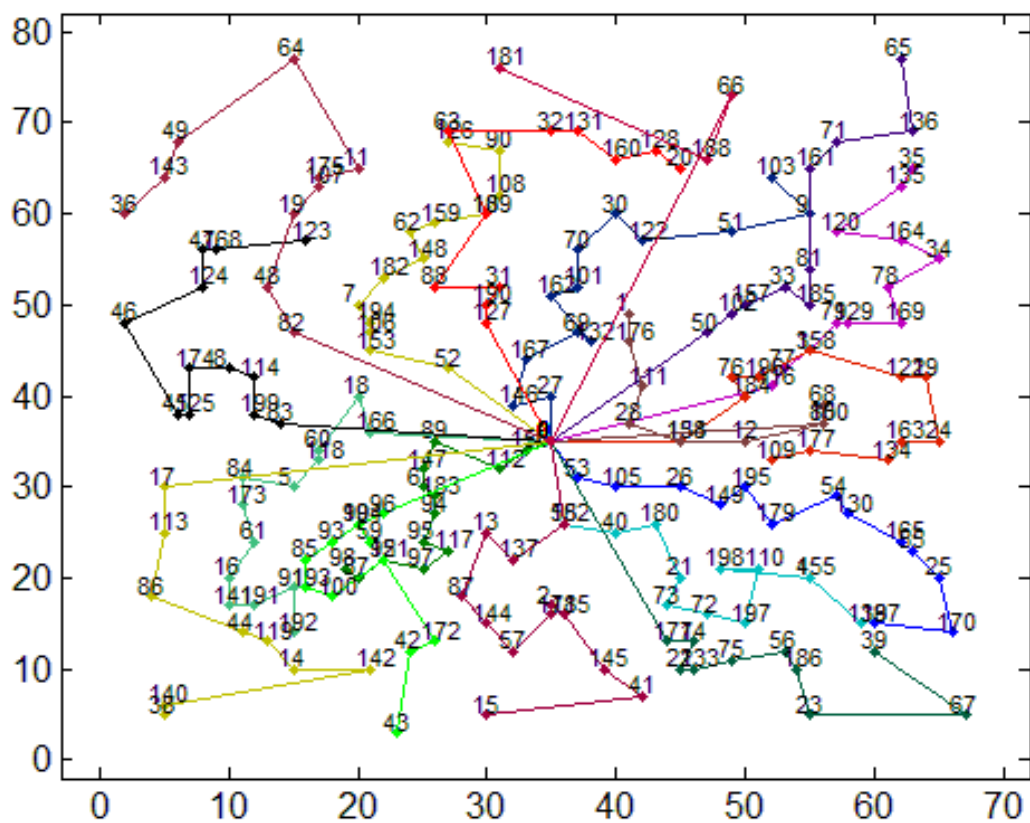
Εφαρμόζοντας τον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών προκύπτει:

- $\text{MinCost}=974,58$
- $K=18$

Η διαδρομή καθενός εκ των 18 οχημάτων:

0	116	77	3	79	129	169	78	34	164	120	135	35	
0	156	112	89	147	6	183	94	95	117	97	92	37	98
0	53	105	26	149	195	179	54	130	165	55	25	170	187
0	27	146	167	69	132	162	101	70	30	122	51	9	103
0	58	40	180	21	73	72	197	110	198	4	139	155	
0	152	137	13	87	144	57	178	2	115	145	41	15	
0	154	184	76	196	158	121	29	24	163	134	177	109	
0	52	153	106	194	7	182	148	62	159	10	108	90	126
0	166	18	60	118	5	84	173	61	16	141	191	91	192
0	127	190	31	88	189	63	32	131	160	128	20		
0	96	99	59	104	93	85	193	100	151	172	42	43	
0	50	102	157	33	185	81	161	71	136	65			
0	83	199	114	8	174	125	45	46	124	47	168	123	
0	82	48	19	107	175	11	64	49	143	36			
0	171	74	22	133	75	56	186	23	67	39			
0	17	113	86	44	119	14	142	140	38				

5.10.2. Γραφική απεικόνιση λύσης παραδείγματος 10



5.11. Παράδειγμα C11

Στο παράδειγμα C11, $n=120$ και $C=200$. Εφαρμόζοντας αρχικά τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor) προκύπτει:

- MinCost=931

5.11.1. Αποτελέσματα εφαρμογής ACO και 2-OPT στο παράδειγμα 11

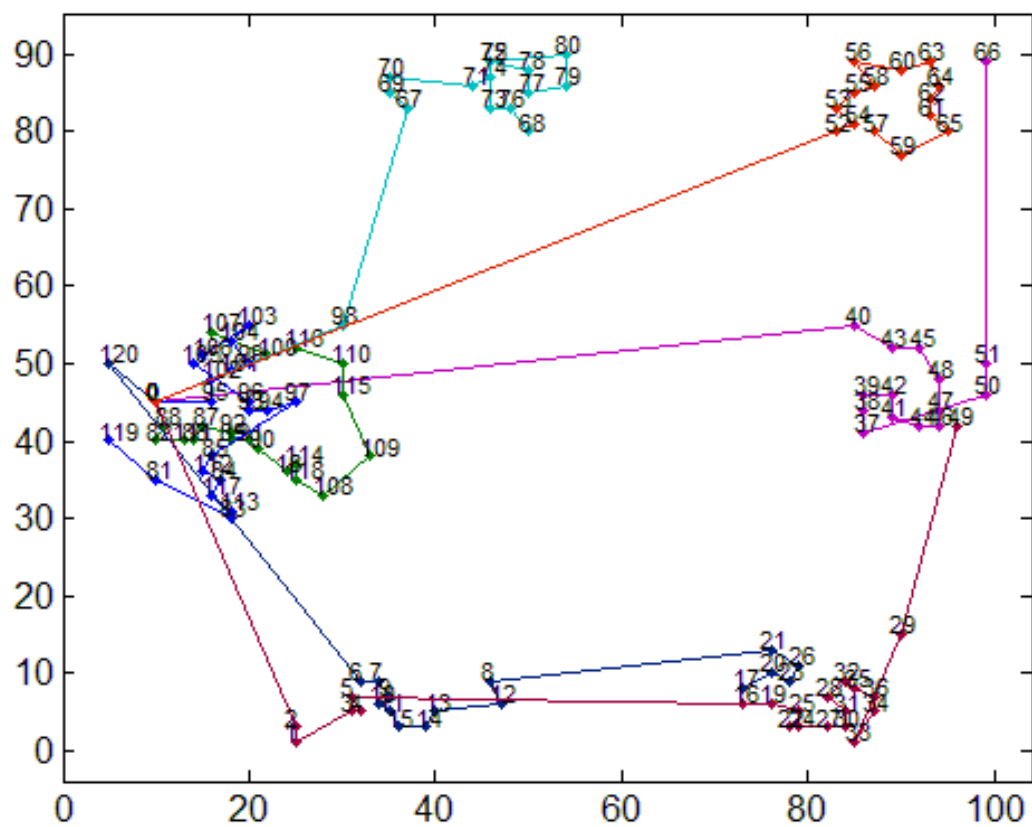
Εφαρμόζοντας τον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών προκύπτει:

- MinCost=837,63
- $K=7$

Η διαδρομή καθενός εκ των 7 οχημάτων:

0	40	43	45	48	47	46	44	41	42	39	38	37	50	51	66						
0	88	82	111	86	87	92	89	91	90	18	114	118	108	109	115	110	116	100	107		
0	95	102	101	99	104	103	106	105	96	93	94	97	85	112	84	117	113	83	81	119	
0	120	6	7	9	10	11	15	14	13	12	8	21	26	23	20	17					
0	98	67	69	70	71	74	72	78	77	79	80	75	73	76	68						
0	2	1	3	4	5	16	19	25	24	22	27	30	31	28	32	35	36	34	33	29	49
0	52	54	53	55	58	56	60	63	64	62	61	65	59	57							

5.11.2. Γραφική απεικόνιση λύσης παραδείγματος 11



5.12. Παράδειγμα C12

Στο παράδειγμα C12, $n=100$ και $C=200$. Εφαρμόζοντας αρχικά τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor) προκύπτει:

- MinCost=760

5.12.1. Αποτελέσματα εφαρμογής ACO και 2-OPT στο παράδειγμα 12

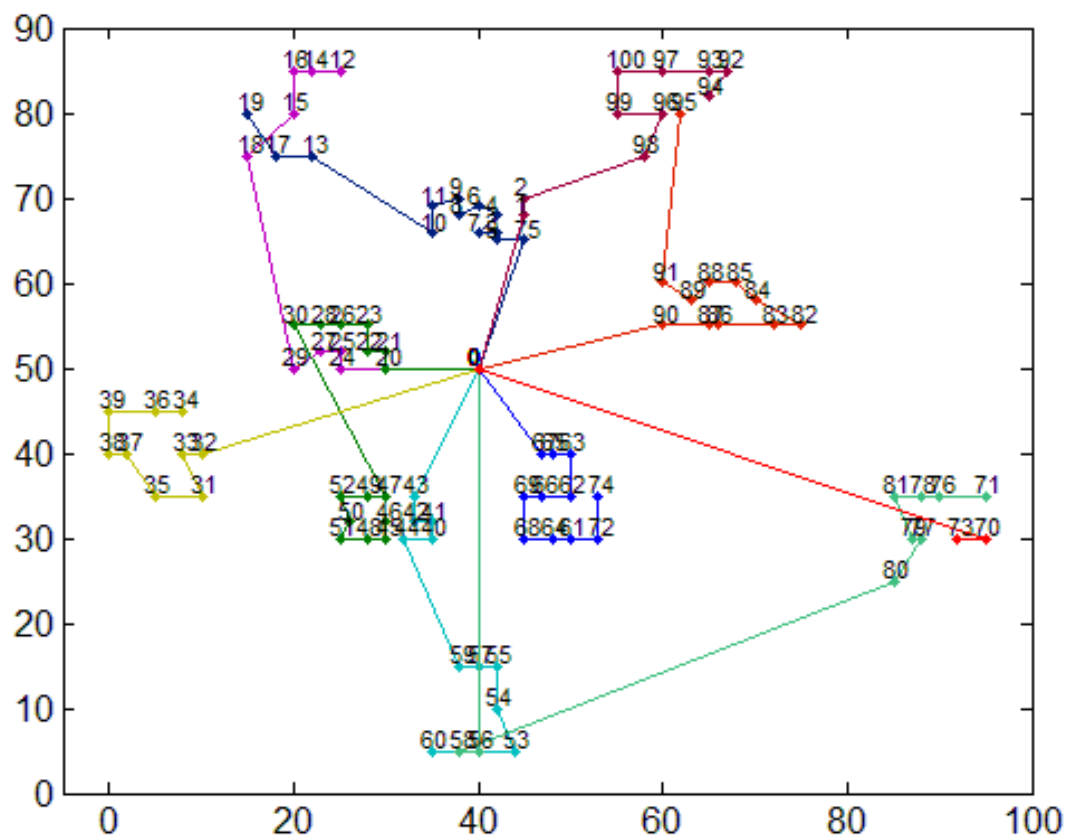
Εφαρμόζοντας τον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών προκύπτει:

- MinCost=688,21
- $K=10$

Η διαδρομή καθενός εκ των 10 οχημάτων:

0	24	25	27	29	18	15	16	14	12						
0	20	21	22	23	26	28	30	47	49	52	50	51	48	45	46
0	67	65	63	62	66	69	68	64	61	72	74				
0	75	5	3	7	4	6	8	9	11	10	13	17	19		
0	43	42	41	40	44	59	57	55	54	53	60				
0	1	2	98	96	99	100	97	93	92	94					
0	90	87	86	83	82	84	85	88	89	91	95				
0	32	33	31	35	37	38	39	36	34						
0	56	58	80	77	79	81	78	76	71						
0	70	73													

5.12.2. Γραφική απεικόνιση λύσης παραδείγματος 12



5.13. Παράδειγμα C13

Στο παράδειγμα C13, $n=120$, $C=200$, $L=648$ και $e=10$. Εφαρμόζοντας αρχικά τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor) προκύπτει:

- $\text{MinCost}=1105$

5.13.1. Αποτελέσματα εφαρμογής ACO και 2-OPT στο παράδειγμα 13

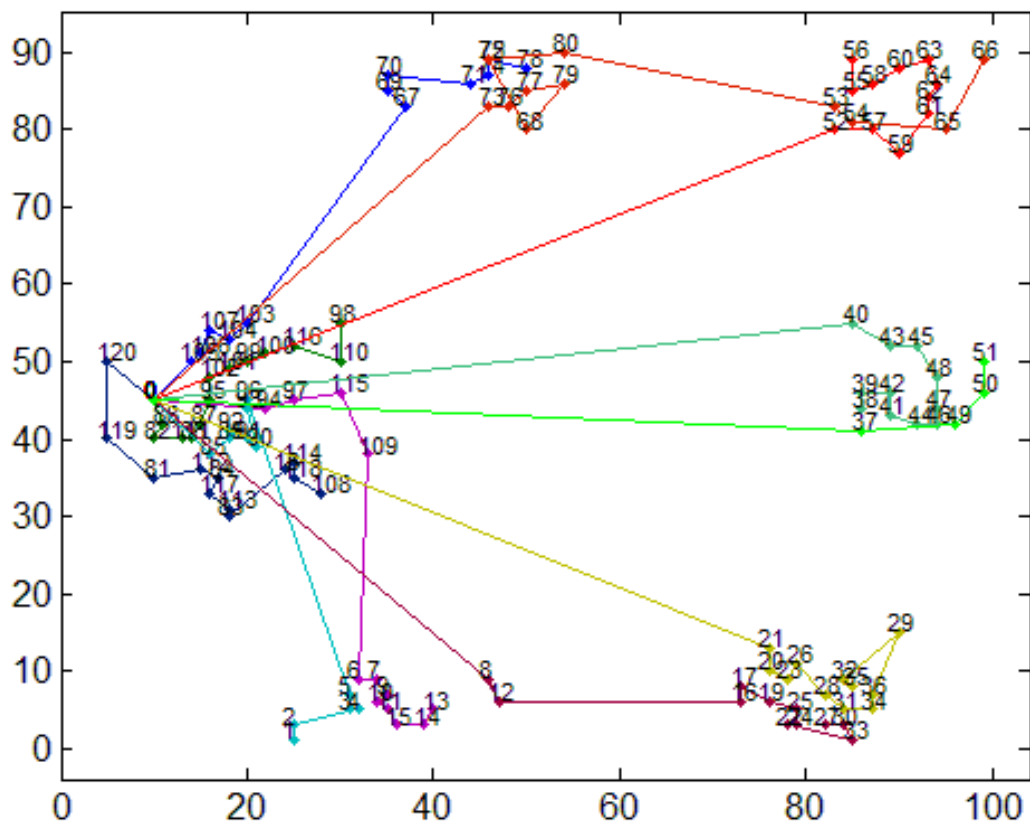
Εφαρμόζοντας τον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών προκύπτει:

- $\text{MinCost}=1048,10$
- $K=11$

Η διαδρομή καθενός εκ των 11 οχημάτων:

0	94	97	115	109	6	7	9	10	11	15	14	13	
0	88	82	111	86	87	95	102	101	99	100	116	110	98
0	105	106	107	104	103	67	69	70	71	74	72	78	
0	120	119	81	112	84	117	113	83	18	114	118	108	
0	85	89	92	91	90	93	96	5	3	4	2	1	
0	8	12	16	17	19	25	24	22	33	30	27		
0	73	76	77	79	68	75	80	53	54	65	66		
0	21	20	23	26	28	31	34	36	29	32	35		
0	40	43	45	48	47	46	44	41	42	39	3		
0	52	57	59	61	62	64	63	60	58	55	56		
0	37	49	50	51									

5.13.2. Γραφική απεικόνιση λύσης παραδείγματος 13



5.14. Παράδειγμα C14

Στο παράδειγμα C14, $n=100$, $C=200$, $L=936$ και $e=10$. Εφαρμόζοντας αρχικά τον αλγόριθμο του πλησιέστερου γείτονα (Nearest Neighbor) προκύπτει:

- $\text{MinCost}=724$

5.14.1. Αποτελέσματα εφαρμογής ACO και 2-OPT στο παράδειγμα 14

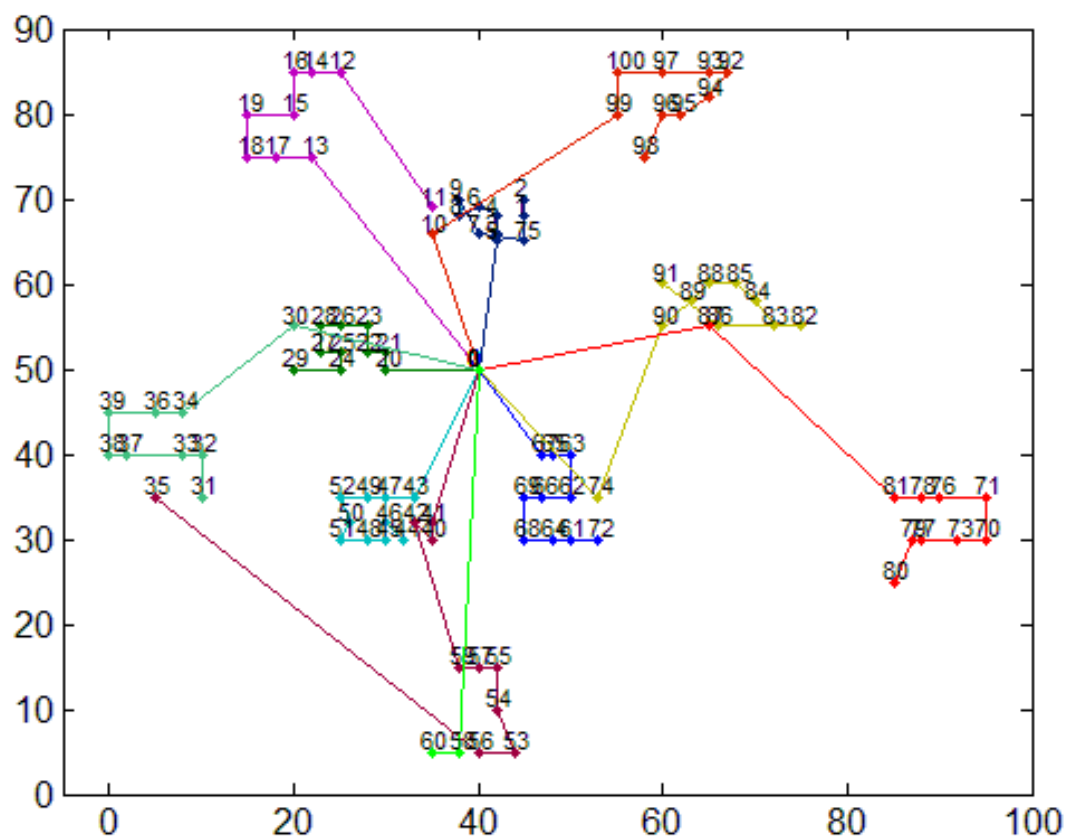
Εφαρμόζοντας τον αλγόριθμο βελτιστοποίησης αποικίας μυρμηγκιών προκύπτει:

- $\text{MinCost}=644,23$
- $K=11$

Η διαδρομή καθενός εκ των 11 οχημάτων:

0	65	63	62	66	69	68	64	61	72	74
0	20	21	22	23	26	28	27	25	24	29
0	67	41	40	42	44	45	46	43	47	49
0	10	11	99	100	97	93	92	94	95	96
0	7	8	9	6	4	3	5	75	1	2
0	30	13	17	18	19	15	16	14	12	
0	90	81	78	76	71	70	73	77	79	80
0	89	88	85	84	83	82	86	87	91	98
0	32	33	34	36	39	38	37	35	31	
0	48	51	50	52						
0	60	58	56	53	54	55	57	59		

5.14.2. Γραφική απεικόνιση λύσης παραδείγματος 14



Κεφάλαιο 6 – Σύνοψη - Σχόλια

6.1. Συνολική παρουσίαση αποτελεσμάτων

Στον πίνακα που ακολουθεί παραθέτονται συνολικά τα αποτελέσματα της εφαρμογής του αλγορίθμου στα 14 προτεινόμενα παραδείγματα. Με έντονη γραφή απεικονίζονται τα κόστη που βρέθηκαν σταδιακά κατά την δημιουργία του αλγορίθμου και φαίνεται η σημαντική βελτίωση του κόστους με την προσθήκη του 2 ορι στον ACO.

Παραδείγματα	Κόστος (βέλτιστο)	Αριθμός Οχημάτων (βέλτιστος)	Κόστος (Nearest Neighbor)	Κόστος (ACO)	Μείωση ACO - Nearest Neighbor	Απόκλιση από βέλτιστο	Αριθμός Οχημάτων (ACO)
C1	416,06	5	492,5	451,20	-8,39%	8,45%	5
C2	567,14	10	810	694,21	-14,30%	22,41%	10
C3	639,74	8	871	811,46	-6,84%	26,84%	8
C4	733,13	12	1018	898,46	-11,74%	22,55%	12
C5	894,11	16	1247	1106,18	-11,29%	23,72%	17
C6	412,96	6	467	457,31	-2,07%	10,74%	6
C7	584,15	10	717	704,23	-1,78%	20,56%	12
C8	644,63	9	765	752,62	-1,62%	16,75%	10
C9	764,56	13	893	878,80	-1,59%	14,94%	14
C10	888,46	17	1060	974,58	-8,06%	9,69%	18
C11	682,12	7	931	837,63	-10,03%	22,80%	7
C12	534,24	10	760	688,21	-9,45%	28,82%	10
C13	910,26	11	1105	1048,10	-5,15%	15,14%	11
C14	591,87	11	724	644,23	-11,02%	8,85%	11

Ο μέσος όρος των αποκλίσεων των λύσεων που παρουσιάζονται, από τις μέχρι στιγμής βέλτιστες, είναι 18,02%.

6.2. Σχολιασμός

Ο αλγόριθμος που αναπτύχθηκε και παρουσιάστηκε σε αυτό το τεύχος, παρήγαγε ικανοποιητικά αποτελέσματα. Η απόκλιση των λύσεων αυτού, από τις λύσεις του HES- hybrid evolution strategy for the open vehicle routing problem που προτείνεται στο [6], είναι μικρές, ενώ σε μερικά παραδείγματα είναι της τάξεως του 8-9%.

Ένα πολύ ενδιαφέρον στοιχείο είναι πως τα αποτελέσματα προέκυψαν με λίγες επαναλήψεις του αλγορίθμου βελτιστοποίησης αποικίας μυρμηγκιών και του 2-opt. Το γεγονός αυτό είναι εξαιρετικά ενθαρρυντικό για περεταίρω ανάλυση και βελτιστοποίηση του αλγορίθμου που προτάθηκε, προσδοκώντας αποτελέσματα με μηδαμινές αποκλίσεις από τα μέχρι στιγμής βέλτιστα, σε παγκόσμια κλίμακα, αποτελέσματα.

Η ταχύτητα υπολογισμού των λύσεων ήταν πολύ ικανοποιητική αν και ο προγραμματισμός έγινε στο περιβάλλον της Matlab, λογισμικό που διδάσκεται στο Πολυτεχνείο Κρήτης, αλλά δεν έχει ιδιαίτερα γρήγορους χρόνους υπολογισμών.

Παράρτημα

Παράδειγμα C1

Κόμβος	Τετμημένη x	Τεταγμένη y	Ζήτηση
1	30	40	0
2	37	52	7
3	49	49	30
4	52	64	16
5	20	26	9
6	40	30	21
7	21	47	15
8	17	63	19
9	31	62	23
10	52	33	11
11	51	21	5
12	42	41	19
13	31	32	29
14	5	25	23
15	12	42	21
16	36	16	10
17	52	41	15
18	27	23	3
19	17	33	41
20	13	13	9
21	57	58	28
22	62	42	8
23	42	57	8
24	16	57	16
25	8	52	10
26	7	38	28
27	27	68	7
28	30	48	15
29	43	67	14
30	58	48	6
31	58	27	19
32	37	69	11
33	38	46	12
34	46	10	23
35	61	33	26
36	62	63	17
37	63	69	6
38	32	22	9
39	45	35	15
40	59	15	14
41	5	6	7
42	10	17	27
43	21	10	13
44	5	64	11
45	30	15	16
46	39	10	10
47	32	39	5
48	25	32	25
49	25	55	17
50	48	28	18
51	56	37	10

Παράδειγμα C2

Κόμβος	Τετμημένη x	Τεταγμένη y	Ζήτηση
1	40	40	0
2	22	22	18
3	36	26	26
4	21	45	11
5	45	35	30
6	55	20	21
7	33	34	19
8	50	50	15
9	55	45	16
10	26	59	29
11	40	66	26
12	55	65	37
13	35	51	16
14	62	35	12
15	62	57	31
16	62	24	8
17	21	36	19
18	33	44	20
19	9	56	13
20	62	48	15
21	66	14	22
22	44	13	28
23	26	13	12
24	11	28	6
25	7	43	27
26	17	64	14
27	41	46	18
28	55	34	17
29	35	16	29
30	52	26	13
31	43	26	22
32	31	76	25
33	22	53	28
34	26	29	27
35	50	40	19
36	55	50	10
37	54	10	12
38	60	15	14
39	47	66	24
40	30	60	16
41	30	50	33
42	12	17	15
43	15	14	11
44	16	19	18
45	21	48	17
46	50	30	21
47	51	42	27
48	50	15	19
49	48	21	20
50	12	38	5
51	15	56	22
52	29	39	12
53	54	38	19

54	55	57	22
55	67	41	16
56	10	70	7
57	6	25	26
58	65	27	14
59	40	60	21
60	70	64	24
61	64	4	13
62	36	6	15
63	30	20	18
64	20	30	11
65	15	5	28
66	50	70	9
67	57	72	37
68	45	42	30
69	38	33	10
70	50	4	8
71	66	8	11
72	59	5	3
73	35	60	1
74	27	24	6
75	40	20	10
76	40	37	20

Παράδειγμα C3

Κόμβος	Τετμημένη x	Τεταγμένη y	Ζήτηση
1	35	35	0
2	41	49	10
3	35	17	7
4	55	45	13
5	55	20	19
6	15	30	26
7	25	30	3
8	20	50	5
9	10	43	9
10	55	60	16
11	30	60	16
12	20	65	12
13	50	35	19
14	30	25	23
15	15	10	20
16	30	5	8
17	10	20	19
18	5	30	2
19	20	40	12
20	15	60	17
21	45	65	9
22	45	20	11
23	45	10	18
24	55	5	29
25	65	35	3
26	65	20	6

27	45	30	17
28	35	40	16
29	41	37	16
30	64	42	9
31	40	60	21
32	31	52	27
33	35	69	23
34	53	52	11
35	65	55	14
36	63	65	8
37	2	60	5
38	20	20	8
39	5	5	16
40	60	12	31
41	40	25	9
42	42	7	5
43	24	12	5
44	23	3	7
45	11	14	18
46	6	38	16
47	2	48	1
48	8	56	27
49	13	52	36
50	6	68	30
51	47	47	13
52	49	58	10
53	27	43	9
54	37	31	14
55	57	29	18
56	63	23	2
57	53	12	6
58	32	12	7
59	36	26	18
60	21	24	28
61	17	34	3
62	12	24	13
63	24	58	19
64	27	69	10
65	15	77	9
66	62	77	20
67	49	73	25
68	67	5	25
69	56	39	36
70	37	47	6
71	37	56	5
72	57	68	15
73	47	16	25
74	44	17	9
75	46	13	8
76	49	11	18
77	49	42	13
78	53	43	14
79	61	52	3
80	57	48	23
81	56	37	6
82	55	54	26

83	15	47	16
84	14	37	11
85	11	31	7
86	16	22	41
87	4	18	35
88	28	18	26
89	26	52	9
90	26	35	15
91	31	67	3
92	15	19	1
93	22	22	2
94	18	24	22
95	26	27	27
96	25	24	20
97	22	27	11
98	25	21	12
99	19	21	10
100	20	26	9
101	18	18	17

Παράδειγμα C4

Κόμβος	Τετμημένη x	Τεταγμένη y	Ζήτηση
1	35	35	0
2	41	49	10
3	35	17	7
4	55	45	13
5	55	20	19
6	15	30	26
7	25	30	3
8	20	50	5
9	10	43	9
10	55	60	16
11	30	60	16
12	20	65	12
13	50	35	19
14	30	25	23
15	15	10	20
16	30	5	8
17	10	20	19
18	5	30	2
19	20	40	12
20	15	60	17
21	45	65	9
22	45	20	11
23	45	10	18
24	55	5	29
25	65	35	3
26	65	20	6
27	45	30	17
28	35	40	16
29	41	37	16
30	64	42	9

31	40	60	21
32	31	52	27
33	35	69	23
34	53	52	11
35	65	55	14
36	63	65	8
37	2	60	5
38	20	20	8
39	5	5	16
40	60	12	31
41	40	25	9
42	42	7	5
43	24	12	5
44	23	3	7
45	11	14	18
46	6	38	16
47	2	48	1
48	8	56	27
49	13	52	36
50	6	68	30
51	47	47	13
52	49	58	10
53	27	43	9
54	37	31	14
55	57	29	18
56	63	23	2
57	53	12	6
58	32	12	7
59	36	26	18
60	21	24	28
61	17	34	3
62	12	24	13
63	24	58	19
64	27	69	10
65	15	77	9
66	62	77	20
67	49	73	25
68	67	5	25
69	56	39	36
70	37	47	6
71	37	56	5
72	57	68	15
73	47	16	25
74	44	17	9
75	46	13	8
76	49	11	18
77	49	42	13
78	53	43	14
79	61	52	3
80	57	48	23
81	56	37	6
82	55	54	26
83	15	47	16
84	14	37	11
85	11	31	7
86	16	22	41

87	4	18	35
88	28	18	26
89	26	52	9
90	26	35	15
91	31	67	3
92	15	19	1
93	22	22	2
94	18	24	22
95	26	27	27
96	25	24	20
97	22	27	11
98	25	21	12
99	19	21	10
100	20	26	9
101	18	18	17
102	37	52	7
103	49	49	30
104	52	64	16
105	20	26	9
106	40	30	21
107	21	47	15
108	17	63	19
109	31	62	23
110	52	33	11
111	51	21	5
112	42	41	19
113	31	32	29
114	5	25	23
115	12	42	21
116	36	16	10
117	52	41	15
118	27	23	3
119	17	33	41
120	13	13	9
121	57	58	28
122	62	42	8
123	42	57	8
124	16	57	16
125	8	52	10
126	7	38	28
127	27	68	7
128	30	48	15
129	43	67	14
130	58	48	6
131	58	27	19
132	37	69	11
133	38	46	12
134	46	10	23
135	61	33	26
136	62	63	17
137	63	69	6
138	32	22	9
139	45	35	15
140	59	15	14
141	5	6	7
142	10	17	27

143	21	10	13
144	5	64	11
145	30	15	16
146	39	10	10
147	32	39	5
148	25	32	25
149	25	55	17
150	48	28	18
151	56	37	10

Παράδειγμα C5

Κόμβος	Τετμημένη x	Τεταγμένη y	Ζήτηση
1	35	35	0
2	41	49	10
3	35	17	7
4	55	45	13
5	55	20	19
6	15	30	26
7	25	30	3
8	20	50	5
9	10	43	9
10	55	60	16
11	30	60	16
12	20	65	12
13	50	35	19
14	30	25	23
15	15	10	20
16	30	5	8
17	10	20	19
18	5	30	2
19	20	40	12
20	15	60	17
21	45	65	9
22	45	20	11
23	45	10	18
24	55	5	29
25	65	35	3
26	65	20	6
27	45	30	17
28	35	40	16
29	41	37	16
30	64	42	9
31	40	60	21
32	31	52	27
33	35	69	23
34	53	52	11
35	65	55	14
36	63	65	8
37	2	60	5
38	20	20	8
39	5	5	16
40	60	12	31

41	40	25	9
42	42	7	5
43	24	12	5
44	23	3	7
45	11	14	18
46	6	38	16
47	2	48	1
48	8	56	27
49	13	52	36
50	6	68	30
51	47	47	13
52	49	58	10
53	27	43	9
54	37	31	14
55	57	29	18
56	63	23	2
57	53	12	6
58	32	12	7
59	36	26	18
60	21	24	28
61	17	34	3
62	12	24	13
63	24	58	19
64	27	69	10
65	15	77	9
66	62	77	20
67	49	73	25
68	67	5	25
69	56	39	36
70	37	47	6
71	37	56	5
72	57	68	15
73	47	16	25
74	44	17	9
75	46	13	8
76	49	11	18
77	49	42	13
78	53	43	14
79	61	52	3
80	57	48	23
81	56	37	6
82	55	54	26
83	15	47	16
84	14	37	11
85	11	31	7
86	16	22	41
87	4	18	35
88	28	18	26
89	26	52	9
90	26	35	15
91	31	67	3
92	15	19	1
93	22	22	2
94	18	24	22
95	26	27	27
96	25	24	20

97	22	27	11
98	25	21	12
99	19	21	10
100	20	26	9
101	18	18	17
102	37	52	7
103	49	49	30
104	52	64	16
105	20	26	9
106	40	30	21
107	21	47	15
108	17	63	19
109	31	62	23
110	52	33	11
111	51	21	5
112	42	41	19
113	31	32	29
114	5	25	23
115	12	42	21
116	36	16	10
117	52	41	15
118	27	23	3
119	17	33	41
120	13	13	9
121	57	58	28
122	62	42	8
123	42	57	8
124	16	57	16
125	8	52	10
126	7	38	28
127	27	68	7
128	30	48	15
129	43	67	14
130	58	48	6
131	58	27	19
132	37	69	11
133	38	46	12
134	46	10	23
135	61	33	26
136	62	63	17
137	63	69	6
138	32	22	9
139	45	35	15
140	59	15	14
141	5	6	7
142	10	17	27
143	21	10	13
144	5	64	11
145	30	15	16
146	39	10	10
147	32	39	5
148	25	32	25
149	25	55	17
150	48	28	18
151	56	37	10
152	22	22	18

153	36	26	26
154	21	45	11
155	45	35	30
156	55	20	21
157	33	34	19
158	50	50	15
159	55	45	16
160	26	59	29
161	40	66	26
162	55	65	37
163	35	51	16
164	62	35	12
165	62	57	31
166	62	24	8
167	21	36	19
168	33	44	20
169	9	56	13
170	62	48	15
171	66	14	22
172	44	13	28
173	26	13	12
174	11	28	6
175	7	43	27
176	17	64	14
177	41	46	18
178	55	34	17
179	35	16	29
180	52	26	13
181	43	26	22
182	31	76	25
183	22	53	28
184	26	29	27
185	50	40	19
186	55	50	10
187	54	10	12
188	60	15	14
189	47	66	24
190	30	60	16
191	30	50	33
192	12	17	15
193	15	14	11
194	16	19	18
195	21	48	17
196	50	30	21
197	51	42	27
198	50	15	19
199	48	21	20
200	12	38	5

Παράδειγμα C6

Κόμβος	Τετμημένη x	Τεταγμένη y	Ζήτηση
1	30	40	0
2	37	52	7
3	49	49	30
4	52	64	16
5	20	26	9
6	40	30	21
7	21	47	15
8	17	63	19
9	31	62	23
10	52	33	11
11	51	21	5
12	42	41	19
13	31	32	29
14	5	25	23
15	12	42	21
16	36	16	10
17	52	41	15
18	27	23	3
19	17	33	41
20	13	13	9
21	57	58	28
22	62	42	8
23	42	57	8
24	16	57	16
25	8	52	10
26	7	38	28
27	27	68	7
28	30	48	15
29	43	67	14
30	58	48	6
31	58	27	19
32	37	69	11
33	38	46	12
34	46	10	23
35	61	33	26
36	62	63	17
37	63	69	6
38	32	22	9
39	45	35	15
40	59	15	14
41	5	6	7
42	10	17	27
43	21	10	13
44	5	64	11
45	30	15	16
46	39	10	10
47	32	39	5
48	25	32	25
49	25	55	17
50	48	28	18
51	56	37	10

Παράδειγμα C7

Κόμβος	Τετμημένη x	Τεταγμένη y	Ζήτηση
1	40	40	0
2	22	22	18
3	36	26	26
4	21	45	11
5	45	35	30
6	55	20	21
7	33	34	19
8	50	50	15
9	55	45	16
10	26	59	29
11	40	66	26
12	55	65	37
13	35	51	16
14	62	35	12
15	62	57	31
16	62	24	8
17	21	36	19
18	33	44	20
19	9	56	13
20	62	48	15
21	66	14	22
22	44	13	28
23	26	13	12
24	11	28	6
25	7	43	27
26	17	64	14
27	41	46	18
28	55	34	17
29	35	16	29
30	52	26	13
31	43	26	22
32	31	76	25
33	22	53	28
34	26	29	27
35	50	40	19
36	55	50	10
37	54	10	12
38	60	15	14
39	47	66	24
40	30	60	16
41	30	50	33
42	12	17	15
43	15	14	11
44	16	19	18
45	21	48	17
46	50	30	21
47	51	42	27
48	50	15	19
49	48	21	20
50	12	38	5
51	15	56	22
52	29	39	12
53	54	38	19

54	55	57	22
55	67	41	16
56	10	70	7
57	6	25	26
58	65	27	14
59	40	60	21
60	70	64	24
61	64	4	13
62	36	6	15
63	30	20	18
64	20	30	11
65	15	5	28
66	50	70	9
67	57	72	37
68	45	42	30
69	38	33	10
70	50	4	8
71	66	8	11
72	59	5	3
73	35	60	1
74	27	24	6
75	40	20	10
76	40	37	20

Παράδειγμα C8

Κόμβος	Τετμημένη x	Τεταγμένη y	Ζήτηση
1	35	35	0
2	41	49	10
3	35	17	7
4	55	45	13
5	55	20	19
6	15	30	26
7	25	30	3
8	20	50	5
9	10	43	9
10	55	60	16
11	30	60	16
12	20	65	12
13	50	35	19
14	30	25	23
15	15	10	20
16	30	5	8
17	10	20	19
18	5	30	2
19	20	40	12
20	15	60	17
21	45	65	9
22	45	20	11
23	45	10	18
24	55	5	29
25	65	35	3
26	65	20	6

27	45	30	17
28	35	40	16
29	41	37	16
30	64	42	9
31	40	60	21
32	31	52	27
33	35	69	23
34	53	52	11
35	65	55	14
36	63	65	8
37	2	60	5
38	20	20	8
39	5	5	16
40	60	12	31
41	40	25	9
42	42	7	5
43	24	12	5
44	23	3	7
45	11	14	18
46	6	38	16
47	2	48	1
48	8	56	27
49	13	52	36
50	6	68	30
51	47	47	13
52	49	58	10
53	27	43	9
54	37	31	14
55	57	29	18
56	63	23	2
57	53	12	6
58	32	12	7
59	36	26	18
60	21	24	28
61	17	34	3
62	12	24	13
63	24	58	19
64	27	69	10
65	15	77	9
66	62	77	20
67	49	73	25
68	67	5	25
69	56	39	36
70	37	47	6
71	37	56	5
72	57	68	15
73	47	16	25
74	44	17	9
75	46	13	8
76	49	11	18
77	49	42	13
78	53	43	14
79	61	52	3
80	57	48	23
81	56	37	6
82	55	54	26

83	15	47	16
84	14	37	11
85	11	31	7
86	16	22	41
87	4	18	35
88	28	18	26
89	26	52	9
90	26	35	15
91	31	67	3
92	15	19	1
93	22	22	2
94	18	24	22
95	26	27	27
96	25	24	20
97	22	27	11
98	25	21	12
99	19	21	10
100	20	26	9
101	18	18	17

Παράδειγμα C9

Κόμβος	Τετμημένη x	Τεταγμένη y	Ζήτηση
1	35	35	0
2	41	49	10
3	35	17	7
4	55	45	13
5	55	20	19
6	15	30	26
7	25	30	3
8	20	50	5
9	10	43	9
10	55	60	16
11	30	60	16
12	20	65	12
13	50	35	19
14	30	25	23
15	15	10	20
16	30	5	8
17	10	20	19
18	5	30	2
19	20	40	12
20	15	60	17
21	45	65	9
22	45	20	11
23	45	10	18
24	55	5	29
25	65	35	3
26	65	20	6
27	45	30	17
28	35	40	16
29	41	37	16
30	64	42	9

31	40	60	21
32	31	52	27
33	35	69	23
34	53	52	11
35	65	55	14
36	63	65	8
37	2	60	5
38	20	20	8
39	5	5	16
40	60	12	31
41	40	25	9
42	42	7	5
43	24	12	5
44	23	3	7
45	11	14	18
46	6	38	16
47	2	48	1
48	8	56	27
49	13	52	36
50	6	68	30
51	47	47	13
52	49	58	10
53	27	43	9
54	37	31	14
55	57	29	18
56	63	23	2
57	53	12	6
58	32	12	7
59	36	26	18
60	21	24	28
61	17	34	3
62	12	24	13
63	24	58	19
64	27	69	10
65	15	77	9
66	62	77	20
67	49	73	25
68	67	5	25
69	56	39	36
70	37	47	6
71	37	56	5
72	57	68	15
73	47	16	25
74	44	17	9
75	46	13	8
76	49	11	18
77	49	42	13
78	53	43	14
79	61	52	3
80	57	48	23
81	56	37	6
82	55	54	26
83	15	47	16
84	14	37	11
85	11	31	7
86	16	22	41

87	4	18	35
88	28	18	26
89	26	52	9
90	26	35	15
91	31	67	3
92	15	19	1
93	22	22	2
94	18	24	22
95	26	27	27
96	25	24	20
97	22	27	11
98	25	21	12
99	19	21	10
100	20	26	9
101	18	18	17
102	37	52	7
103	49	49	30
104	52	64	16
105	20	26	9
106	40	30	21
107	21	47	15
108	17	63	19
109	31	62	23
110	52	33	11
111	51	21	5
112	42	41	19
113	31	32	29
114	5	25	23
115	12	42	21
116	36	16	10
117	52	41	15
118	27	23	3
119	17	33	41
120	13	13	9
121	57	58	28
122	62	42	8
123	42	57	8
124	16	57	16
125	8	52	10
126	7	38	28
127	27	68	7
128	30	48	15
129	43	67	14
130	58	48	6
131	58	27	19
132	37	69	11
133	38	46	12
134	46	10	23
135	61	33	26
136	62	63	17
137	63	69	6
138	32	22	9
139	45	35	15
140	59	15	14
141	5	6	7
142	10	17	27

143	21	10	13
144	5	64	11
145	30	15	16
146	39	10	10
147	32	39	5
148	25	32	25
149	25	55	17
150	48	28	18
151	56	37	10

Παράδειγμα C10

Κόμβος	Τετμημένη x	Τεταγμένη y	Ζήτηση
1	35	35	0
2	41	49	10
3	35	17	7
4	55	45	13
5	55	20	19
6	15	30	26
7	25	30	3
8	20	50	5
9	10	43	9
10	55	60	16
11	30	60	16
12	20	65	12
13	50	35	19
14	30	25	23
15	15	10	20
16	30	5	8
17	10	20	19
18	5	30	2
19	20	40	12
20	15	60	17
21	45	65	9
22	45	20	11
23	45	10	18
24	55	5	29
25	65	35	3
26	65	20	6
27	45	30	17
28	35	40	16
29	41	37	16
30	64	42	9
31	40	60	21
32	31	52	27
33	35	69	23
34	53	52	11
35	65	55	14
36	63	65	8
37	2	60	5
38	20	20	8
39	5	5	16
40	60	12	31

41	40	25	9
42	42	7	5
43	24	12	5
44	23	3	7
45	11	14	18
46	6	38	16
47	2	48	1
48	8	56	27
49	13	52	36
50	6	68	30
51	47	47	13
52	49	58	10
53	27	43	9
54	37	31	14
55	57	29	18
56	63	23	2
57	53	12	6
58	32	12	7
59	36	26	18
60	21	24	28
61	17	34	3
62	12	24	13
63	24	58	19
64	27	69	10
65	15	77	9
66	62	77	20
67	49	73	25
68	67	5	25
69	56	39	36
70	37	47	6
71	37	56	5
72	57	68	15
73	47	16	25
74	44	17	9
75	46	13	8
76	49	11	18
77	49	42	13
78	53	43	14
79	61	52	3
80	57	48	23
81	56	37	6
82	55	54	26
83	15	47	16
84	14	37	11
85	11	31	7
86	16	22	41
87	4	18	35
88	28	18	26
89	26	52	9
90	26	35	15
91	31	67	3
92	15	19	1
93	22	22	2
94	18	24	22
95	26	27	27
96	25	24	20

97	22	27	11
98	25	21	12
99	19	21	10
100	20	26	9
101	18	18	17
102	37	52	7
103	49	49	30
104	52	64	16
105	20	26	9
106	40	30	21
107	21	47	15
108	17	63	19
109	31	62	23
110	52	33	11
111	51	21	5
112	42	41	19
113	31	32	29
114	5	25	23
115	12	42	21
116	36	16	10
117	52	41	15
118	27	23	3
119	17	33	41
120	13	13	9
121	57	58	28
122	62	42	8
123	42	57	8
124	16	57	16
125	8	52	10
126	7	38	28
127	27	68	7
128	30	48	15
129	43	67	14
130	58	48	6
131	58	27	19
132	37	69	11
133	38	46	12
134	46	10	23
135	61	33	26
136	62	63	17
137	63	69	6
138	32	22	9
139	45	35	15
140	59	15	14
141	5	6	7
142	10	17	27
143	21	10	13
144	5	64	11
145	30	15	16
146	39	10	10
147	32	39	5
148	25	32	25
149	25	55	17
150	48	28	18
151	56	37	10
152	22	22	18

153	36	26	26
154	21	45	11
155	45	35	30
156	55	20	21
157	33	34	19
158	50	50	15
159	55	45	16
160	26	59	29
161	40	66	26
162	55	65	37
163	35	51	16
164	62	35	12
165	62	57	31
166	62	24	8
167	21	36	19
168	33	44	20
169	9	56	13
170	62	48	15
171	66	14	22
172	44	13	28
173	26	13	12
174	11	28	6
175	7	43	27
176	17	64	14
177	41	46	18
178	55	34	17
179	35	16	29
180	52	26	13
181	43	26	22
182	31	76	25
183	22	53	28
184	26	29	27
185	50	40	19
186	55	50	10
187	54	10	12
188	60	15	14
189	47	66	24
190	30	60	16
191	30	50	33
192	12	17	15
193	15	14	11
194	16	19	18
195	21	48	17
196	50	30	21
197	51	42	27
198	50	15	19
199	48	21	20
200	12	38	5

Παράδειγμα C11

Κόμβος	Τετμημένη x	Τεταγμένη y	Ζήτηση
1	10	45	0
2	25	1	25
3	25	3	7
4	31	5	13
5	32	5	6
6	31	7	14
7	32	9	5
8	34	9	11
9	46	9	19
10	35	7	5
11	34	6	15
12	35	5	15
13	47	6	17
14	40	5	13
15	39	3	12
16	36	3	18
17	73	6	13
18	73	8	18
19	24	36	12
20	76	6	17
21	76	10	4
22	76	13	7
23	78	3	12
24	78	9	13
25	79	3	8
26	79	5	16
27	79	11	15
28	82	3	6
29	82	7	5
30	90	15	9
31	84	3	11
32	84	5	10
33	84	9	3
34	85	1	7
35	87	5	2
36	85	8	4
37	87	7	4
38	86	41	18
39	86	44	14
40	86	46	12
41	85	55	17
42	89	43	20
43	89	46	14
44	89	52	16
45	92	42	10
46	92	52	9
47	94	42	11
48	94	44	7
49	94	48	13
50	96	42	5
51	99	46	4
52	99	50	21
53	83	80	13

54	83	83	11
55	85	81	12
56	85	85	14
57	85	89	10
58	87	80	8
59	87	86	16
60	90	77	19
61	90	88	5
62	93	82	17
63	93	84	7
64	93	89	16
65	94	86	14
66	95	80	17
67	99	89	13
68	37	83	17
69	50	80	13
70	35	85	14
71	35	87	16
72	44	86	7
73	46	89	13
74	46	83	9
75	46	87	11
76	46	89	35
77	48	83	5
78	50	85	28
79	50	88	7
80	54	86	3
81	54	90	10
82	10	35	7
83	10	40	12
84	18	30	11
85	17	35	10
86	16	38	8
87	14	40	11
88	15	42	21
89	11	42	4
90	18	40	15
91	21	39	16
92	20	40	4
93	18	41	16
94	20	44	7
95	22	44	10
96	16	45	9
97	20	45	11
98	25	45	17
99	30	55	12
100	20	50	11
101	22	51	7
102	18	49	9
103	16	48	11
104	20	55	12
105	18	53	7
106	14	50	8
107	15	51	6
108	16	54	5
109	28	33	12

110	33	38	13
111	30	50	7
112	13	40	7
113	15	36	8
114	18	31	11
115	25	37	13
116	30	46	11
117	25	52	10
118	16	33	7
119	25	35	4
120	5	40	20
121	5	50	13

Παράδειγμα C12

Κόμβος	Τετμημένη x	Τεταγμένη y	Ζήτηση
1	40	50	0
2	45	68	10
3	45	70	30
4	42	66	10
5	42	68	10
6	42	65	10
7	40	69	20
8	40	66	20
9	38	68	20
10	38	70	10
11	35	66	10
12	35	69	10
13	25	85	20
14	22	75	30
15	22	85	10
16	20	80	40
17	20	85	40
18	18	75	20
19	15	75	20
20	15	80	10
21	30	50	10
22	30	52	20
23	28	52	20
24	28	55	10
25	25	50	10
26	25	52	40
27	25	55	10
28	23	52	10
29	23	55	20
30	20	50	10
31	20	55	10
32	10	35	20
33	10	40	30
34	8	40	40
35	8	45	20
36	5	35	10
37	5	45	10

38	2	40	20
39	0	40	30
40	0	45	20
41	35	30	10
42	35	32	10
43	33	32	20
44	33	35	10
45	32	30	10
46	30	30	10
47	30	32	30
48	30	35	10
49	28	30	10
50	28	35	10
51	26	32	10
52	25	30	10
53	25	35	10
54	44	5	20
55	42	10	40
56	42	15	10
57	40	5	30
58	40	15	40
59	38	5	30
60	38	15	10
61	35	5	20
62	50	30	10
63	50	35	20
64	50	40	50
65	48	30	10
66	48	40	10
67	47	35	10
68	47	40	10
69	45	30	10
70	45	35	10
71	95	30	30
72	95	35	20
73	53	30	10
74	92	30	10
75	53	35	50
76	45	65	20
77	90	35	10
78	88	30	10
79	88	35	20
80	87	30	10
81	85	25	10
82	85	35	30
83	75	55	20
84	72	55	10
85	70	58	20
86	68	60	30
87	66	55	10
88	65	55	20
89	65	60	30
90	63	58	10
91	60	55	10
92	60	60	10
93	67	85	20

94	65	85	40
95	65	82	10
96	62	80	30
97	60	80	10
98	60	85	30
99	58	75	20
100	55	80	10
101	55	85	20

Παράδειγμα C13

Κόμβος	Τετμημένη x	Τεταγμένη y	Ζήτηση
1	10	45	0
2	25	1	25
3	25	3	7
4	31	5	13
5	32	5	6
6	31	7	14
7	32	9	5
8	34	9	11
9	46	9	19
10	35	7	5
11	34	6	15
12	35	5	15
13	47	6	17
14	40	5	13
15	39	3	12
16	36	3	18
17	73	6	13
18	73	8	18
19	24	36	12
20	76	6	17
21	76	10	4
22	76	13	7
23	78	3	12
24	78	9	13
25	79	3	8
26	79	5	16
27	79	11	15
28	82	3	6
29	82	7	5
30	90	15	9
31	84	3	11
32	84	5	10
33	84	9	3
34	85	1	7
35	87	5	2
36	85	8	4
37	87	7	4
38	86	41	18
39	86	44	14
40	86	46	12
41	85	55	17

42	89	43	20
43	89	46	14
44	89	52	16
45	92	42	10
46	92	52	9
47	94	42	11
48	94	44	7
49	94	48	13
50	96	42	5
51	99	46	4
52	99	50	21
53	83	80	13
54	83	83	11
55	85	81	12
56	85	85	14
57	85	89	10
58	87	80	8
59	87	86	16
60	90	77	19
61	90	88	5
62	93	82	17
63	93	84	7
64	93	89	16
65	94	86	14
66	95	80	17
67	99	89	13
68	37	83	17
69	50	80	13
70	35	85	14
71	35	87	16
72	44	86	7
73	46	89	13
74	46	83	9
75	46	87	11
76	46	89	35
77	48	83	5
78	50	85	28
79	50	88	7
80	54	86	3
81	54	90	10
82	10	35	7
83	10	40	12
84	18	30	11
85	17	35	10
86	16	38	8
87	14	40	11
88	15	42	21
89	11	42	4
90	18	40	15
91	21	39	16
92	20	40	4
93	18	41	16
94	20	44	7
95	22	44	10
96	16	45	9
97	20	45	11

98	25	45	17
99	30	55	12
100	20	50	11
101	22	51	7
102	18	49	9
103	16	48	11
104	20	55	12
105	18	53	7
106	14	50	8
107	15	51	6
108	16	54	5
109	28	33	12
110	33	38	13
111	30	50	7
112	13	40	7
113	15	36	8
114	18	31	11
115	25	37	13
116	30	46	11
117	25	52	10
118	16	33	7
119	25	35	4
120	5	40	20
121	5	50	13

Παράδειγμα C14

Κόμβος	Τετμημένη x	Τεταγμένη y	Ζήτηση
1	40	50	0
2	45	68	10
3	45	70	30
4	42	66	10
5	42	68	10
6	42	65	10
7	40	69	20
8	40	66	20
9	38	68	20
10	38	70	10
11	35	66	10
12	35	69	10
13	25	85	20
14	22	75	30
15	22	85	10
16	20	80	40
17	20	85	40
18	18	75	20
19	15	75	20
20	15	80	10
21	30	50	10
22	30	52	20
23	28	52	20
24	28	55	10
25	25	50	10

26	25	52	40
27	25	55	10
28	23	52	10
29	23	55	20
30	20	50	10
31	20	55	10
32	10	35	20
33	10	40	30
34	8	40	40
35	8	45	20
36	5	35	10
37	5	45	10
38	2	40	20
39	0	40	30
40	0	45	20
41	35	30	10
42	35	32	10
43	33	32	20
44	33	35	10
45	32	30	10
46	30	30	10
47	30	32	30
48	30	35	10
49	28	30	10
50	28	35	10
51	26	32	10
52	25	30	10
53	25	35	10
54	44	5	20
55	42	10	40
56	42	15	10
57	40	5	30
58	40	15	40
59	38	5	30
60	38	15	10
61	35	5	20
62	50	30	10
63	50	35	20
64	50	40	50
65	48	30	10
66	48	40	10
67	47	35	10
68	47	40	10
69	45	30	10
70	45	35	10
71	95	30	30
72	95	35	20
73	53	30	10
74	92	30	10
75	53	35	50
76	45	65	20
77	90	35	10
78	88	30	10
79	88	35	20
80	87	30	10
81	85	25	10

82	85	35	30
83	75	55	20
84	72	55	10
85	70	58	20
86	68	60	30
87	66	55	10
88	65	55	20
89	65	60	30
90	63	58	10
91	60	55	10
92	60	60	10
93	67	85	20
94	65	85	40
95	65	82	10
96	62	80	30
97	60	80	10
98	60	85	30
99	58	75	20
100	55	80	10
101	55	85	20

Βιβλιογραφία

- [1] Ιωάννης Μαρινάκης, Αθανάσιος Μυγδαλάς. Σχεδιασμός και Βελτιστοποίηση Εφοδιαστικής Αλυσίδας. Εκδόσεις “σοφία”, Θεσσαλονίκη 2008
- [2] Martin Christopher, Logistics and Supply Chain Management Creating Value - Adding Networks Third Edition, Pearson Education 2005
- [3] Fisher, M. L., Jaikumar, R., and Wassenhove, L. N. V. (1986). A Multiplier Adjustment Method for the Generalized Assignment Problem, Management Science, 32 (9), 1095-1103
- [4] Bodin L, Golden B, Assad A, Ball M. Routing and scheduling of vehicles and crews: the state of the art. Computers & Operations Research 1983;10:63-211
- [5] Levy L. Private Communication. RouteSmart Technologies, Inc., 2005
- [6] Repoussis P.P., Tarantilis C.D., Braysy O. and Ioannou G., “A hybrid evolution strategy for the open vehicle routing problem”. Computers & Operations Research, forthcoming (doi:10.1016/j.cor.2008.11.003)
- [7] Li F, Golden B and Wasil E. The open vehicle routing problem: algorithms, large-scale test problems, and computational results. Computers & Operations Research 2004;34(10):2918-30
- [8] Emmanouil E. Zachariadis and Chris T. Kianoudis. An open vehicle routing problem metaheuristic for examining wide solution neighborhoods. Computers & Operations Research, forthcoming (doi:10.1016/j.cor.2009.06.021)
- [9] Tarantilis CD, Diakoulaki D, Kirianoudis CT. Combination of geographical information system and effective routing algorithms for real life distribution operations, Eur J Res 2004;152:437-53
- [10] Tarantilis CD, Kirianoudis CT, Ioannou D, Prastacos GP. Solving the open vehicle routing problem via a single parameter metaheuristic algorithm. J Oper Res Soc 2005;56:588-96.
- [11] Tarantilis CD, Ioannou G, Kirianoudis CT, Prastacos GP. A threshold accepting approach to the open vehicle routing problem. RAIRO 2004;38:345-60