



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ ΟΜΑΔΑΣ ΜΕ ΧΩΡΗΤΙΚΟΤΗΤΑ ΜΕ ΧΡΗΣΗ ΑΛΓΟΡΙΘΜΟΥ ΜΕΤΑΒΛΗΤΗΣ ΓΕΙΤΟΝΙΑΣ ΑΝΑΖΗΤΗΣΗΣ

Solving Capacitated Team Orienteering Problem with Variable Neighborhood Search Algorithm



Γεώργιος Κουλίδης

Επιβλέπων καθηγητής: Δρ. Ιωάννης Μαρινάκης

Χανιά 2017

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε υπό την επίβλεψη του επίκουρου καθηγητή κ. Ιωάννη Μαρινάκη, στο πλαίσιο του πενταετούς προγράμματος σπουδών της σχολής Μηχανικών Παραγωγής και Διοίκησης.

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον κ. Ιωάννη Μαρινάκη για το ενδιαφέρον θέμα διπλωματικής εργασίας που μου ανέθεσε, καθώς και για την ουσιαστική καθοδήγηση και συμβολή του στην υλοποίηση της εργασίας.

Εν συνέχεια θα ήθελα να ευχαριστήσω τους συμφοιτητές και φίλους μου για την ηθική υποστήριξη και την ψυχαγωγία που μου πρόσφεραν όλα αυτά τα χρόνια .

Κλείνοντας, θα ήθελα να ευχαριστήσω τους γονείς μου, για την αμέριστη υποστήριξη και κατανόηση που έχουν δείξει καθ' όλη τη διάρκεια της πενταετούς μου φοίτησης στο Πολυτεχνείο Κρήτης, καθώς και για τις θυσίες που έχουν κάνει όλα αυτά τα χρόνια.

Περιεχόμενα

Ευχαριστίες	2
Περίληψη	4
ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ.....	5
1.1 Η εφοδιαστική Αλυσίδα (Supply Chain)	5
1.2 Εφοδιαστική (Logistics).....	5
1.3 Μεταφορές και Αποθέματα	6
ΚΕΦΑΛΑΙΟ 2: ΠΡΟΒΛΗΜΑ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ ΟΜΑΔΑΣ ΜΕ ΧΩΡΗΤΙΚΟΤΗΤΑ (CTOP)	7
2.1 Εισαγωγή	7
2.2 Μοντελοποίηση.....	7
2.3 Μαθηματική απεικόνιση	7
ΚΕΦΑΛΑΙΟ 3: ΜΕΘΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ	9
3.1 Εισαγωγή	9
3.2 Αλγόριθμος Μεταβλητής Γειτονιάς Αναζήτησης (Variable Neighborhood Search).....	10
3.3 Άλλοι μεθευρετικοί αλγόριθμοι	11
ΚΕΦΑΛΑΙΟ 4: ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ ΟΜΑΔΑΣ ΜΕ ΧΩΡΗΤΙΚΟΤΗΤΑ	14
4.1 Εισαγωγή	14
4.2 Περιγραφή και μοντελοποίηση του προβλήματος.....	14
4.3 Εύρεση αρχικής λύσης με τον πλησιέστερο γείτονα	15
4.3.3 Αρχική λύση για το Πρόβλημα Προσανατολισμού Ομάδας με Χωρητικότητα (CTOP)	16
4.4 Εύρεση της βέλτιστης λύσης με τη χρήση του αλγορίθμου μεταβλητής γειτονιάς αναζήτησης.....	21
4.4.1 Εφαρμογή τοπικής αναζήτησης 2-1 exchange(2-1 ανταλλαγή).....	22
4.4.2 Εφαρμογή τοπικής αναζήτησης 1-0 relocate(1-0 επανατοποθέτηση)	24
4.5 Τερματισμός αλγορίθμου και ψευτοκώδικας.....	25
ΚΕΦΑΛΑΙΟ 5: Περιγραφή και αναπαράσταση αποτελεσμάτων για το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα (CTOP)	28
5.1 Γενική περιγραφή των αποτελεσμάτων	28
5.2 Πίνακας αποτελεσμάτων.....	39
ΚΕΦΑΛΑΙΟ 6: Σύγκριση αποτελεσμάτων και συμπεράσματα	40
6.1 Σύγκριση αποτελεσμάτων.....	40
6.2 Συμπεράσματα	40
Βιβλιογραφία	42

Περίληψη

Λόγω της παγκοσμιοποίησης και των διαρκών αυξανόμενων απαιτήσεων των πελατών, η ανάπτυξη της εφοδιαστικής αλυσίδας αποτελεί καθοριστικό ρόλο για τις επιχειρήσεις. Οι βασικές δραστηριότητες της Εφοδιαστικής Αλυσίδας είναι οι μεταφορές και τα αποθέματα τα οποία απορροφούν μεγάλο μερίδιο του κόστους. Συνεπώς ο σκοπός της διαχείρισης της Εφοδιαστικής Αλυσίδας είναι η ελαχιστοποίηση αυτού του κόστους. Στη παρούσα διπλωματική εργασία επιλύουμε το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα (CTOP). Στο πρόβλημα μας έχουμε πελάτες οι οποίοι δεν μπορούν να εξυπηρετηθούν πάνω από 1 φορά. Κάθε πελάτης έχει ντετερμινιστική ζήτηση και κέρδος. Το όχημα μπορεί να συλλέξει το κέρδος από τον πελάτη ικανοποιώντας του την ζήτηση, αλλά η συνολική ζήτηση από όλους του πελάτες δεν θα πρέπει να ξεπερνάει την συνολική χωρητικότητα του οχήματος και ένα συγκεκριμένο μήκος διαδρομής. Η διαδρομή ολοκληρώνεται όταν ένα όχημα δεν μπορεί να εξυπηρετήσει άλλον πελάτη. Στόχος του προβλήματος είναι η μεγιστοποίηση του συνολικού κέρδους που έχει συλλεχθεί από τα οχήματα [1]. Αρχικά υλοποιείται ο αλγόριθμος του Πλησιέστερου Γείτονα όπου και προκύπτουν οι αρχικές εφικτές μας λύσεις. Στη συνέχεια εφαρμόζεται ο Αλγόριθμος Μεταβλητής Γειτονίας Αναζήτησης (Variable Neighborhood Search Algorithm) με σκοπό την βελτίωση της λύσης. Στη εργασία παρουσιάζεται η υλοποίηση του VNS αλγορίθμου καθώς και τα αποτελέσματα από τη χρήση του αλγορίθμου. Για την ανάπτυξη του αλγορίθμου χρησιμοποιήθηκε το προγραμματιστικό περιβάλλον Matlab.

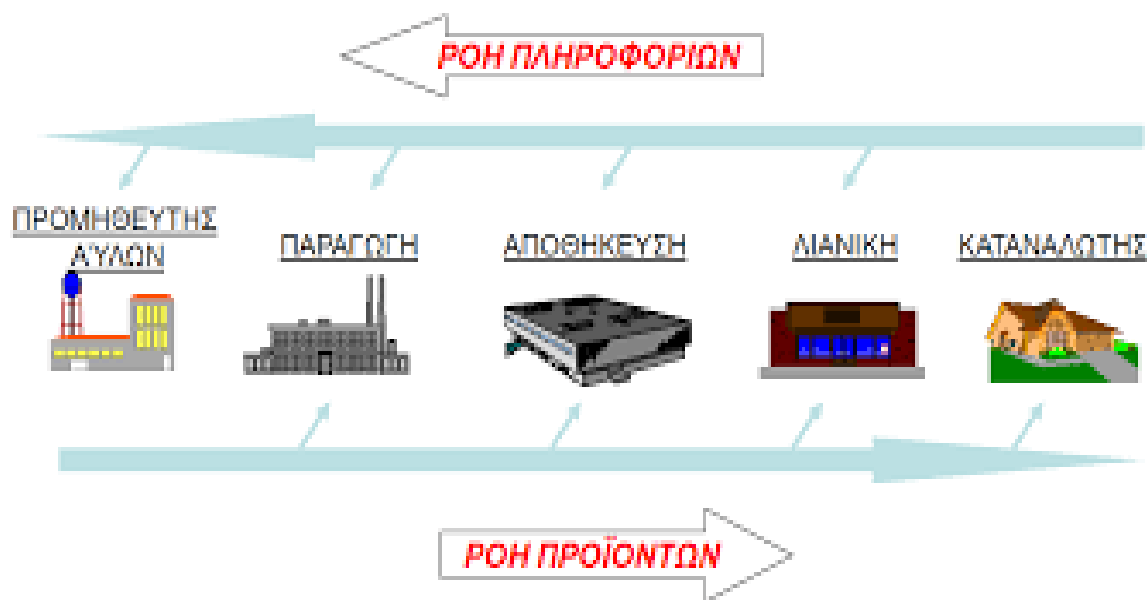
ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1.1 Η εφοδιαστική Αλυσίδα (Supply Chain)

Στις μέρες μας λόγω της παγκοσμιοποίησης και της συνεχούς εξέλιξης της τεχνολογίας όλο και περισσότερες επιχειρήσεις αρχίζουν να στηρίζονται στην εφοδιαστική αλυσίδα προκειμένου να ανταγωνιστούν την παγκόσμια αγορά. Στόχος της εφοδιαστικής αλυσίδας είναι να καλύψει τις απαιτήσεις των πελατών και να εξασφαλίσει αύξηση τόσο των μεριδίων αγοράς όσο και την κερδοφορία της επιχείρησης .

Με τον όρο εφοδιαστική αλυσίδα (supply chain) εννοούμε την ροή υλικών, πληροφοριών και υπηρεσιών από τους προμηθευτές μέσα από τα εργοστάσια και τις αποθήκες, στους τελικούς πελάτες [2]. Επομένως, η Εφοδιαστική Αλυσίδα αποτελείται από κατασκευαστές , προμηθευτές, τους χώρους αποθήκευσης, κέντρα διανομών, μεταφορείς, τους πελάτες, τους πωλητές, τις πρώτες ύλες, τα αποθέματα που παράγονται από το σύστημα παραγωγής αλλά και από τα έτοιμα προϊόντα που μπορεί να υπάρχουν ανάμεσα. Η εφοδιαστική αλυσίδα είναι δυναμικό σύστημα που εξελίσσεται με τον χρόνο και εμπλέκει διάφορες δραστηριότητες της αλυσίδας.

Στην εφοδιαστική αλυσίδα όλα τα στάδια δημιουργούν δαπάνες. Συνεπώς, βασική μας επιδίωξη σε κάθε εφοδιαστική αλυσίδα είναι να μείνει ικανοποιημένος ο πελάτης αφού αποτελεί την μοναδική πηγή εσόδων.



Εικόνα 1.1: Εφοδιαστική Αλυσίδα

1.2 Εφοδιαστική (Logistics)

Οι πρώτες ιστορικές αναφορές εμφανίστηκαν από την εποχή που ξεκίνησαν οι κατακτητικοί πόλεμοι. Με τον σωστό σχεδιασμό οι ηγέτες εξασφάλιζαν έναν πιο δυνατό στρατό με καλύτερη οργάνωση, τόσο στο πεδίο της μάχης όσο και στις μετακινήσεις τους. Επίσης, ιστορικές αναφορές για την εφοδιαστική εμφανίστηκαν και στην αρχαία Αίγυπτο όπου ήταν απαραίτητος ο σωστός σχεδιασμός για την αποθήκευση και την μεταφορά μεγάλης ποσότητας υλικών για την κατασκευή των πυραμίδων.

Η εφοδιαστική είναι η διαδικασία που σχεδιάζει, υλοποιεί και ελέγχει την αποδοτική και αποτελεσματική ροή και αποθήκευση των προϊόντων, υπηρεσιών και σχετικών πληροφοριών από το σημείο προέλευσης τους έως το σημείο κατανάλωσής τους, ώστε να ικανοποιούν οι απαιτήσεις των πελατών [3]. Εν γένει, είναι το τμήμα που έχει σαν στόχο να φτάσει το επιθυμητό τελικό προϊόν, στην επιθυμητή ποσότητα, την χρονική στιγμή και στο σημείο που επιθυμεί ο πελάτης.

1.3 Μεταφορές και Αποθέματα

Η δραστηριότητα της μεταφοράς και της διανομής αφορά τον τρόπο και τα μέσα για την μεταβίβαση υλικών μέσω των φυσικών καναλιών. Για τη μεταφορά των υλικών πρέπει να γίνει η κατάλληλη επιλογή μέσου, τύπου οχήματος, ο καθορισμός των διαδρομών και των δρομολογήσεων, των μεταφορών κ.λπ. Οι μεταφορές αποτελούν την πιο δαπανηρή δραστηριότητα καθώς καταλαμβάνουν το μεγαλύτερο κομμάτι της πίτας εξόδων της εφοδιαστικής αλυσίδας. Εξαιτίας του μεγάλου κόστους οι επιχειρήσεις δίνουν μεγάλη βαρύτητα ώστε να εξασφαλίσουν την ομαλή λειτουργία της μεταφοράς

Συνήθεις προβλήματα που δημιουργούνται στο σχεδιασμό των μεταφορών είναι η επιλογή του στόλου μεταφοράς (μέγεθος του στόλου και χρήση διαφορετικών τύπων οχημάτων), τη δρομολόγηση των οχημάτων (λαμβάνοντας υπόψιν τους περιορισμούς μέγιστης χωρητικότητας και το μέγιστο μήκος διαδρομής), το σχεδιασμό του δικτύου παραγωγής (βελτίωση δρομολογίων και ενδιάμεσων αποθηκών) και την κατάλληλη επιλογή του προσωπικού που θα εκτελέσει τις διανομές.

ΚΕΦΑΛΑΙΟ 2: ΠΡΟΒΛΗΜΑ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ ΟΜΑΔΑΣ ΜΕ ΧΩΡΗΤΙΚΟΤΗΤΑ (CTOP)

2.1 Εισαγωγή

Το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα (CTOP) είναι παραλλαγή του προβλήματος δρομολόγησης οχημάτων (Vehicle Routing Problem). Στο CTOP δεν μπορούν να ικανοποιηθούν όλοι οι πελάτες λόγω των περιορισμών που διέπουν το πρόβλημα. Σε αυτό το πρόβλημα κάθε πελάτης έχει ένα κέρδος το οποίο συλλέγεται όταν ικανοποιηθεί ο πελάτης. Επίσης, κάθε πελάτης έχει και μια συγκεκριμένη ζήτηση. Κάθε όχημα αρχίζει και τελειώνει σε ένα συγκεκριμένο μέρος, την αποθήκη. Το όχημα δεν θα πρέπει να υπερβαίνει τους περιορισμούς της ζήτησης και του μέγιστου μήκους διαδρομής. Στόχος του προβλήματος είναι η μεγιστοποίηση του κέρδους που θα συλλεχθεί από τους πελάτες [4].

2.2 Μοντελοποίηση

Θεωρούμε το πρόβλημα CTOP ορισμένο σε ένα γράφημα $G=(V, E)$, όπου $V=\{1, \dots, n\}$ είναι το σύνολο των κόμβων του γραφήματος και E είναι το σύνολο των τόξων. Η αποθήκη αντιστοιχεί στον κόμβο 0. Το τόξο $(i,j) \in E$ παρουσιάζει μια πιθανή μετάβαση, να ταξιδέψουμε από τον κόμβο i στον κόμβο j . Ένα μη αρνητικό κέρδος και μια μη αρνητική ζήτηση συσχετίζονται με κάθε τόξο (i,j) . Κάθε τόξο $(i,j) \in E$ συσχετίζεται με ένα κόστος C_{ij} και αντιπροσωπεύει το κόστος ταξιδιού από τον κόμβο i στον κόμβο j .

Άλλοι παράμετροι του προβλήματος είναι:

d_i η ζήτηση του κόμβου i

m ο αριθμός των ομοιογενών οχημάτων

Q η χωρητικότητα του οχήματος

T το μήκος που μπορεί να διανύσει κάθε όχημα

P το κέρδος των πελατών

Ο σκοπός του προβλήματος CTOP είναι η εύρεση των δρομολογίων κατά τέτοιο τρόπο ώστε να εξυπηρετούνται οι κατάλληλοι πελάτες που θα μας αποφέρουν το μεγαλύτερο κέρδος. Ως δρομολόγιο ορίζεται ένα κύκλωμα που αρχίζει και τελειώνει στην αποθήκη με την συνολική ζήτηση και το συνολικό μήκος να μην ξεπερνάει την μέγιστη χωρητικότητα Q και το μέγιστο μήκος του οχήματος αντίστοιχα.

2.3 Μαθηματική απεικόνιση

Ας ορίσουμε $\Omega = \{r_1, \dots, r_{|\Omega|}\}$ τις πιθανές διαδρομές για το όχημα. Η μεταβλητή $a_{ik}=1$ όταν η διαδρομή $r_k \in \Omega$ επισκέπτεται τον πελάτη i , ενώ $a_{ik}=0$ όταν η διαδρομή $r_k \in \Omega$ δεν επισκέπτεται τον πελάτη i . Η μεταβλητή c_k είναι το συνολικό κέρδος που έχει η διαδρομή $r_k \in \Omega$: $c_k = \sum_{i \in V} a_{ik} p_i$.

Το μαθηματικό μοντέλο του CTOP [5] είναι:

Maximize

$$\sum_{r_k \in \Omega} c_k x_k \quad (1)$$

Υπό τους περιορισμούς

$$\sum_{r_k \in \Omega} a_{ik} x_k \leq 1, i \in V \setminus \{1\} \quad (2)$$

$$\sum_{r_k \in \Omega} x_k \leq m \quad (3)$$

$$x_k \in \{0,1\}, r_k \in \Omega \quad (4)$$

Η δυαδική μεταβλητή x_k υποδηλώνει εάν μια διαδρομή $r_k \in \Omega$ χρησιμοποιείται ή όχι. Ο περιορισμός (2) εξασφαλίζει ότι κάθε πελάτης επισκέπτεται μια φορά. Ο περιορισμός (3) περιορίζει τον αριθμό των οχημάτων που χρησιμοποιούνται σε m .

ΚΕΦΑΛΑΙΟ 3: ΜΕΘΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

3.1 Εισαγωγή

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. Οι μεθευρετικοί αλγόριθμοι μπορούν να χωριστούν σε διάφορες κατηγορίες. Ένας καλός διαχωρισμός των μεθευρετικών αλγορίθμων είναι ανάλογα με το πόσες λύσεις χρησιμοποιούν. Υπάρχουν αλγόριθμοι που χρησιμοποιούν μια λύση και κάνουν αναζήτηση στη γειτονία της λύσης και υπάρχουν αλγόριθμοι που έχουν έναν πληθυσμό από λύσεις και προσπαθούν να κάνουν αναζήτηση σε όλο το χώρο λύσεων. Τέλος υπάρχουν και υβριδικές μορφές αυτών των δυο κατηγοριών.

Τα δύο κύρια χαρακτηριστικά κάθε μεθευρετικού αλγορίθμου είναι η επίταση (intensification) και η διαφοροποίηση (diversification). Η επίταση σημαίνει η εστίαση στην τοπική αναζήτηση εκμεταλλευόμενη την πληροφορία ότι βρίσκουμε μια καλή λύση σε αυτή την περιοχή. Από την άλλη μεριά η διαφοροποίηση σημαίνει η εξερεύνηση του χώρου των λύσεων προς αναζήτηση νέας βέλτιστης λύσης.

Οι μεθευρετικοί αλγόριθμοι που επικεντρώνονται στην αναζήτηση γύρω από κάποιο σημείο που έχει βρεθεί το τοπικό ελάχιστο με τη μέθοδο της τοπικής αναζήτησης μπορούν να διακριθούν σε τέσσερις βασικές κατηγορίες, ανάλογα με τον τρόπο που χρησιμοποιούν για να αποφύγουν το τοπικό ελάχιστο.

1. Επαναληπτικές διαδικασίες που αρχίζουν από διαφορετικές αρχικές λύσεις. Οι πιο χαρακτηριστικές μέθοδοι αυτής της κατηγορίας είναι :
 - Η μέθοδος της διαδικασίας Άπληστης Τυχαιοποιημένης Προσαρμοστικής Αναζήτησης (Greedy Randomized Adaptive Search Procedure – GRASP)
 - Αλγόριθμοι Πολυενακτικής Τοπικής Αναζήτησης (Multistart Local Search)
 - Αλγόριθμοι της Επαναληπτικής Τοπικής Αναζήτησης (Iterated Local Search)
2. Αλγόριθμοι που δέχονται γειτονικές κινήσεις που δεν βελτιώνουν τη λύση. Με αυτό τον τρόπο μπορεί σε μια από τις επόμενες κινήσεις να ξεφύγουμε από το τοπικό ελάχιστο και να οδηγηθούμε σε κάποιο επόμενο τοπικό ελάχιστο το οποίο να είναι καλύτερο από το τρέχων τοπικό ελάχιστο. Οι δύο πιο χαρακτηριστικοί αλγόριθμοι που εκφράζουν αυτή την κατηγορία είναι :
 - Προσομοιωμένη Ανόπτηση (Simulated Annealing)
 - Περιορισμένη Αναζήτηση (Tabu Search)
3. Αλγόριθμοι που αλλάζουν τη γειτονία αναζήτησης. Αυτή η κατηγορία αποτελείται από αλγορίθμους οι οποίοι αν κολλήσουν σε κάποιο τοπικό ελάχιστο αλλάζουν τον αλγόριθμο

που χρησιμοποιούν για την αναζήτηση σε γειτονικά σημεία του χώρου λύσεων. Οι πιο χαρακτηριστικές μέθοδοι αυτής της κατηγορίας είναι :

- Αλγόριθμος Μεταβλητής Γειτονιάς Αναζήτησης (Variable Neighborhood Search – VNS)
 - Αλγόριθμος Επέκτασης της Γειτονιάς Αναζήτησης (Expanding Neighborhood Search – ENS)
4. Αλγόριθμοι που αλλάζουν την αντικειμενική συνάρτηση ή κάποια από τα δεδομένα του προβλήματος. Η πιο χαρακτηριστική μέθοδος αυτής της κατηγορίας είναι ο :
- Αλγόριθμος Καθοδηγούμενης Τοπικής Αναζήτησης (Guided Local Search)

3.2 Αλγόριθμος Μεταβλητής Γειτονιάς Αναζήτησης (Variable Neighborhood Search)

Στην παρούσα διπλωματική εργασία δουλέψαμε με τον μεθευρετικό αλγόριθμο Μεταβλητής Γειτονιάς Αναζήτησης (VNS), ούτως ώστε να βρούμε καλύτερη λύση από την αρχική μας λύση που προκύπτει από την εφαρμογή του αλγορίθμου του πλησιέστερου γείτονα.

Ο αλγόριθμος της μεταβλητής γειτονιάς αναζήτησης εφευρέθηκε από τους Pierre Hansen και Nenad Mladenovic και παρουσιάστηκε πρώτη φορά το 1997 [6] [7]. Η βασική ιδέα είναι η χρήση πολλών μεθόδων τοπικής αναζήτησης για να βρεθεί μια καλύτερη λύση ή για να ξεφύγει ο αλγόριθμος από κάποιο τοπικό ελάχιστο.

Ο αλγόριθμος μεταβλητής γειτονιάς είναι ένας στοχαστικός αλγόριθμος, ο οποίος επιλέγει ένα σύνολο γειτονιών (N_l , όπου $l = 1, \dots, l_{\max}$). Στη συνέχεια η κάθε επανάληψη του αλγορίθμου ακολουθεί τα εξής βήματα : την ανακίνηση (Shaking), την τοπική αναζήτηση (Local Search) και την κίνηση (Move). Στην ουσία σε κάθε επανάληψη μια αρχική λύση s' δημιουργείται από την τρέχουσα γειτονιά αναζήτησης (με τυχαίο ή συστηματικό τρόπο). Εφαρμόζεται τοπική αναζήτηση στην λύση s' με στόχο να παράγουμε τη λύση s'' . Έπειτα ελέγχεται η αντικειμενική συνάρτηση και αν η λύση οδηγεί σε βελτίωση του κόστους τότε η καινούργια λύση αντικαθιστά την αρχική λύση. Η διαδικασία αυτή αρχίζει από την αρχή με τη χρήση της γειτονιάς N_1 και τη καινούργια λύση s'' που βρήκαμε προηγουμένως. Αν δεν βρεθεί μια καλύτερη λύση τότε ο αλγόριθμος προχωρά στην επόμενη γειτονιά αναζήτησης. Παρακάτω δίνεται ένας ψευδοκώδικας του αλγορίθμου γειτονιάς αναζήτησης.

Αλγόριθμος Μεταβλητή Γειτονιά Αναζήτησης

Αρχικοποίηση

Επέλεξε ένα σύνολο γειτονιών (N_l , όπου $l = 1, \dots, l_{\max}$)

Επέλεξε μια αρχική λύση s_0

$l=1$

repeat

Δημιούργησε μία λύση s' στη γειτονιά του N_i

$s'' = LS(s)$, Εφάρμοσε μία διαδικασία τοπικής αναζήτησης στο s'

if $f(s'') < f(s')$ **then**

$s = s''$

$l = 1$

else

$l = l + 1$

end if

until $l \leq l_{\max}$

Επέστρεψε τη βέλτιστη λύση

3.3 Άλλοι μεθευρετικοί αλγόριθμοι

Διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης (Greedy Randomized Adaptive Search Procedure – GRASP)

Η διαδικασία άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης είναι μια επαναληπτική διαδικασία για την εύρεση προσεγγιστικών λύσεων σε προβλήματα συνδυαστικής βελτιστοποίησης [8]. Αυτή η τεχνική μας παρέχει μια εφικτή λύση σε κάθε επανάληψη. Οι επαναλήψεις της διαδικασίας GRASP σταματούν όταν κάποιο κριτήριο τερματισμού ικανοποιείται. Το τελικό αποτέλεσμα είναι η καλύτερη λύση που βρέθηκε σε όλες τις επαναλήψεις. Σε κάθε επανάληψη έχουμε 2 φάσεις και είναι οι εξής : Μια φάση κατασκευής αρχικής λύσης (construction phase) και μια διαδικασία τοπικής αναζήτησης (Local Search phase) για βελτιστοποίηση αυτής της λύσης. Στην πρώτη φάση χρησιμοποιείται μια τυχαιοποιημένη συνάρτηση απληστίας για να κατασκευαστεί η αρχική λύση. Στη συνέχεια αυτή η λύση βελτιώνεται με τη χρήση της διαδικασίας τοπικής αναζήτησης. Η στρατηγική επιλογής του επόμενου στοιχείου βασίζεται στην τυχαία επιλογή από μια λίστα υποψηφίων, που ονομάζεται λίστα περιορισμού των υποψηφίων για εισαγωγή στη λύση την οποία κάθε στοιχείο κατατάσσεται βάσει μιας συνάρτησης απληστίας.

Προσομοιωμένη Ανόπτηση (Simulated Annealing)

Ο πρώτος που πρότεινε την χρησιμοποίηση της προσομοιωμένης απόπτωσης ως μέθοδο για την επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης ήταν ο Kirkpatrick [9]. Η ονομασία του αλγορίθμου προέρχεται από την αναλογία μεταξύ στην προσομοίωση της απόπτωσης υλικών και την στρατηγική επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης. Στην μεταλλουργία η απόπτωση είναι η θερμική κατεργασία όπου ένα υλικό θερμαίνεται μέχρι το σημείο τήξης του και στη συνέχεια ψύχεται αργά. Όταν σταματήσει η ψύξη έχουμε το κατώτερο επίπεδο ενέργειας. Ο αλγόριθμος της προσομοιωμένης απόπτωσης προσομοιώνει τις διάφορες εναλλαγές της ενέργειας που γίνονται σε ένα σύστημα μέχρι να συγκλίνει σε ένα σημείο ισορροπίας. Αυτό στην συνδυαστική βελτιστοποίηση σημαίνει ότι όταν ο αλγόριθμος σταματήσει την κίνηση στο εφικτό χώρο αναζήτησης τότε έχει εντοπίσει το τελικό σημείο ισορροπίας, δηλαδή έχει βρει μια εφικτή λύση.

Αλγόριθμος επαναληπτικής τοπικής αναζήτησης (Iterated Local Search – ILS)

Στον αλγόριθμο επαναληπτικής τοπικής αναζήτησης γίνεται μια προσπάθεια να βελτιωθούν διαδοχικά τοπικά ελάχιστα. Η μέθοδος της επαναληπτικής τοπικής αναζήτησης βελτιώνει τις κλασσικές μεθόδους πολυενακτήριας τοπικής αναζήτησης εισάγοντας στη λύση μια διαταραχή που οδηγεί σε τοπικό ελάχιστο. Η νέα λύση αποθηκεύεται γιατί υπάρχει περίπτωση να χαθεί. Ο αλγόριθμος αποτελείται από 3 βασικά συστατικά :

- 1. Αλγόριθμος τοπικής αναζήτησης .** Ο αλγόριθμος έχει την δυνατότητα να χρησιμοποιεί διαφορετικές μεθόδους σε κάθε επανάληψη υπό την προϋπόθεση να μην χειροτερεύει ο υπολογιστικός φόρτος.
- 2. Την διαδικασία διαταραχής.** Η διαδικασία διαταραχής κρατάει υποχρεωτικά κάποια στοιχεία της αρχικής λύσης, έτσι ώστε να μην είναι τυχαία η καινούργια λύση, σε συνδυασμό με την μετάλλαξη των υπόλοιπων στοιχείων ελπίζοντας να μεταφερθεί σε άλλο σημείο που μπορεί να οδηγήσει σε καλύτερο τοπικό ελάχιστο ή και στο βέλτιστο.
- 3. Το κριτήριο της αποδοχής.** Το κριτήριο αποδοχής περιγράφει τις προϋποθέσεις κάτω από τις οποίες το νέο τοπικό ελάχιστο μπορεί να αντικαταστήσει το παλιό τοπικό ελάχιστο. Η λύση που θα επιλεγεί θα πρέπει να έχει την ικανότητα εντατικοποίησης του αλγορίθμου γύρω από κάποιο σημείο και στη συνέχεια την ικανότητα του αλγορίθμου να μπορεί να εξερευνήσει όσο το δυνατό περισσότερα μέρη του χώρου λύσεων.

Περιορισμένη Αναζήτηση (Tabu Search – TS)

Ο μεθευρετικός αλγόριθμος της περιορισμένης αναζήτησης είναι ο πιο γνωστός μεθευρετικός αλγόριθμος και παρουσιάστηκε από τον Glover [10], [11]. Η περιορισμένη αναζήτηση χρησιμοποιεί έναν ευρετικό αλγόριθμο για να μετακινηθεί από την μία λύση στην άλλη. Ωστόσο υπάρχει ο κίνδυνος η λύση να παγιδευτεί σε τοπικό ελάχιστο, όπως συμβαίνει στους περισσότερους μεθευρετικούς αλγορίθμους. Για μην εγκλωβιστεί η λύση σε τοπικό ελάχιστο χρησιμοποιείται μια συγκεκριμένη στρατηγική για την επιλογή της επόμενης λύσης. Η φιλοσοφία της στρατηγικής αυτής είναι η χρησιμοποίηση μνήμης για τις προηγούμενες πραγματοποιηθέντες κινήσεις, ώστε να ξεφύγει από το τοπικό ελάχιστο. Για να μην γίνεται επανάληψη των λύσεων, οι κινήσεις που γίνονται καταγράφονται σε μία λίστα, η οποία ονομάζεται λίστα περιορισμένων κινήσεων (tabu list) και οι συγκεκριμένες κινήσεις απαγορεύεται να επιστρέψουν στην λύση για ένα συγκεκριμένο αριθμό επαναλήψεων.

Αλγόριθμος Επανασύνδεσης Διαδρομών (Path Relinking – PR)

Ο αλγόριθμος της επανασύνδεσης διαδρομών δημιουργεί καινούργιες λύσεις εξετάζοντας διαφορετικές τροχιές που συνδέουν υψηλής ποιότητας λύσεις. Ο αλγόριθμος ξεκινάει από μια από αυτές τις λύσεις που ονομάζεται εναρκτήρια λύση (starting solution) και καταλήγει στη λύση στόχου ή τελική λύση (target solution). Η βασική ιδέα του αλγορίθμου είναι να δημιουργηθούν και να εξερευνηθούν τροχιές στο χώρο που συνδέουν την αρχική με την τελική λύση. Ο βασικός στόχος του αλγορίθμου είναι η απόσταση μεταξύ της αρχικής και της τελικής λύσης στο τέλος των επαναλήψεων να είναι μηδέν, δηλαδή να είναι ίδιες.

ΚΕΦΑΛΑΙΟ 4: ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΠΡΟΣΑΝΑΤΟΛΙΣΜΟΥ ΟΜΑΔΑΣ ΜΕ ΧΩΡΗΤΙΚΟΤΗΤΑ

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα παρουσιαστούν αναλυτικά τα χαρακτηριστικά των πελατών και των οχημάτων καθώς επίσης και οι περιορισμοί του προβλήματος. Επιπλέον, θα περιγραφεί η μοντελοποίηση του προβλήματος και η αναπαράσταση του στο προγραμματιστικό περιβάλλον Matlab. Τέλος, αναλύονται τα βήματα του Αλγορίθμου Μεταβλητής Γειτονίας Αναζήτησης (VNS).

4.2 Περιγραφή και μοντελοποίηση του προβλήματος

Στην παρούσα διπλωματική εργασία ασχολούμαστε με την επίλυση του προβλήματος προσανατολισμού ομάδας με χωρητικότητα (Capacitated Team Orienteering Problem/CTOP). Στο συγκεκριμένο πρόβλημα, όπως αναφέρθηκε και στα προηγούμενα κεφάλαια, διαθέτουμε μια αποθήκη, ένα συγκεκριμένο αριθμό πελατών με συγκεκριμένη ζήτηση και συγκεκριμένο κέρδος, και ένα στόλο οχημάτων ίδιας και πεπερασμένης χωρητικότητας το καθένα. Η ιδιαιτερότητα του προβλήματος είναι ότι δεν μπορούμε να εξυπηρετήσουμε όλους τους πελάτες διότι, δεν επαρκούν τα οχήματα μας. Ο στόχος του προβλήματος CTOP είναι η εύρεση των δρομολογίων κατά τέτοιο τρόπο ώστε να εξυπηρετούνται οι κατάλληλοι πελάτες που θα μας αποφέρουν το μεγαλύτερο κέρδος ικανοποιώντας του περιορισμούς. Σε πρώτη φάση, θα βρεθεί μια αρχική υποβέλτιστη εφικτή λύση χρησιμοποιώντας των αλγόριθμο του πλησιέστερου γείτονα. Στη συνέχεια έχοντας ως δεδομένη αυτή την αρχική λύση, καλούμαστε να τη βελτιστοποιήσουμε με τη χρήση του μεθευρετικού αλγορίθμου μεταβλητής γειτονιάς αναζήτησης. Οι αλγόριθμοι επιλύθηκαν στο προγραμματιστικό περιβάλλον της Matlab έχοντας γνωστά δεδομένα εισόδου.

Αρχικά, πρέπει να μελετήσουμε το πρόβλημα και να δούμε ποιες μεταβλητές πρέπει να αρχικοποιήσουμε και ποιες να κατασκευάσουμε στη συνέχεια. Ένα από τα δεδομένα που έχουμε είναι η αρίθμηση των πελατών, με τον πελάτη 0 να είναι η αποθήκη. Όμως στο προγραμματιστικό περιβάλλον της Matlab, η αποθήκη αναπαρίσταται με τον αριθμό 1, ο πελάτης 1 με τον αριθμό 2, κ.ο.κ. Τα υπόλοιπα δεδομένα είναι οι συντεταγμένες x και y , η ζήτηση και το κέρδος του κάθε πελάτη, και ο αριθμός των οχημάτων. Τα δεδομένα αυτά αποθηκεύτηκαν στο excel ως αρχεία .csv και διαβάστηκαν από την Matlab με την εντολή xlsread. Για παράδειγμα το κέρδος αποθηκεύτηκε σε έναν μονοδιάστατο πίνακα “P” και η εντολή που διάβαζε τα δεδομένα ήταν $P=xlsread('Profit.xlsx')$. Αντίστοιχα αποθηκεύτηκαν και τα υπόλοιπα δεδομένα του προβλήματος. Αναλυτικά τα δεδομένα παρουσιάζονται παρακάτω:

N: Ο αριθμός των πελατών

d: Διάνυσμα με την ζήτηση του κάθε πελάτη

Coordinates: Πίνακας (N,2) με τις συντεταγμένες x, y του κάθε πελάτη

P: Διάνυσμα με το κέρδος του κάθε πελάτη

Q_{\max} : Η χωρητικότητα του οχήματος

T_{\max} : Το μέγιστο μήκος διαδρομής

Στη συνέχεια, δημιουργήσαμε κάποιους επιπλέον πίνακες και μεταβλητές που μας είναι απαραίτητοι για να επιλύσουμε το πρόβλημα. Ο πρώτος πίνακας που δημιουργήσαμε υπολογίζει τις αποστάσεις μεταξύ όλων των κόμβων. Ο πίνακας συμβολίζεται ως $T(i,j)$, όπου i είναι ο προηγούμενος πελάτης και όπου j ο επόμενος πελάτης. Η απόσταση που υπολογίζεται είναι ευκλείδεια και ο πίνακας είναι συμμετρικός, που σημαίνει ότι η απόσταση για να μεταβούμε από τον πελάτη i στον πελάτη j , ισούται με την απόσταση για να μεταβούμε από τον πελάτη j στον πελάτη i . Σημαντικό είναι να μην ληφθεί υπόψιν η απόσταση του πελάτη από τον εαυτό του. Για το λόγο αυτό τα σημεία $T(i,i)$ στον πίνακα απειρίζονται. Η διάσταση του πίνακα T είναι $N \times N$, δηλαδή όσο είναι και το πλήθος των κόμβων (πελάτες και αποθήκη μαζί). Ο πίνακας κατασκευάστηκε ως εξής.

$$T(i,j) = \sqrt{(Coordinates(i,1) - Coordinates(j,1))^2 + (Coordinates(i,2) - Coordinates(j,2))^2}$$

Για λόγους διευκόλυνσης θεωρούμε ότι υπάρχει ισοδυναμία στις μονάδες χρόνου και απόστασης, δηλαδή ο χρόνος που θα χρειαστεί για να μεταβούμε από τον πελάτη i στον πελάτη j ισούται με την απόσταση που θα διανύσουμε.

4.3 Εύρεση αρχικής λύσης με τον πλησιέστερο γείτονα

Μετά την αποθήκευση των δεδομένων το επόμενο βήμα είναι η υλοποίηση ενός απλού αλγορίθμου απληστίας για την εύρεση μια αρχικής εφικτής λύσης. Αρχικά, γίνεται αρχικοποίηση των μεταβλητών και των πινάκων που θα χρησιμοποιηθούν για την επίλυση του προβλήματος. Τη χρονική στιγμή $t=0$ γνωρίζουμε ότι ο στόλος των οχημάτων βρίσκεται στην αποθήκη με πλήρη διαθέσιμη χωρητικότητα. Επομένως ορίζουμε τις μεταβλητές:

$L=1$: Ο αριθμός των χρησιμοποιούμενων οχημάτων, ορίζεται ως 1 γιατί αναφερόμαστε στην χρονική στιγμή που θα ξεκινήσει το πρώτο όχημα.

$Diadromi=zeros(N,N)$: Αρχικοποίηση του πίνακα που θα περιέχει τις εφικτές διαδρομές.

$Diadromi(L,1)=1$: Η κάθε διαδρομή ξεκινάει από την αποθήκη.

$Apostasis=zeros(N,N)$: Κρατάει όλες τις αποστάσεις της διαδρομής.

$Kostos=zeros(N,1)$: Αρχικοποίηση του πίνακα που θα περιέχει το συνολικό κόστος της διαδρομής.

$Zitisi=zeros(N,1)$: Αρχικοποίηση του πίνακα που θα περιέχει την συνολική ζήτηση της διαδρομής.

$Q=0$: Η τρέχουσα χωρητικότητα του οχήματος. Κάνουμε την παραδοχή ότι το φορτηγό ξεκινάει άδειο και γεμίζει σταδιακά ανάλογα με την ζήτηση των πελατών.

$K=0$: Το τρέχων κόστος, δηλαδή η τρέχουσα απόσταση που έχει διανύσει το όχημα.

$P_{diadromwn}=0$: Το τρέχων κέρδος που έχει συλλεχθεί από τους πελάτες.

4.3.3 Αρχική λύση για το Πρόβλημα Προσανατολισμού Ομάδας με Χωρητικότητα (CTOP)

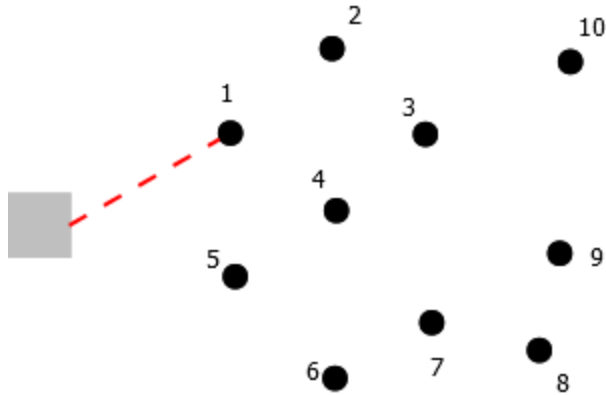
Ο αλγόριθμος του πλησιέστερου γείτονα σε κάθε επανάληψη αναζητεί καλύτερη τοπική λύση. Σε κάθε επανάληψη εξετάζει και έναν πελάτη τον οποίο συνυπολογίζει στην διαδρομή και αν ικανοποιούνται οι περιορισμοί τον κρατάει στο διάνυσμα της διαδρομής. Ο αλγόριθμος επαναλαμβάνεται μέχρις ότου να μην υπάρχουν διαθέσιμα φορτηγά στην αποθήκη. Οι περιορισμοί του προβλήματος είναι:

- Κάθε πελάτης πρέπει να εξυπηρετείται μόνο μια φορά, από ένα και μόνο φορτηγό.
- Το σύνολο της ζήτησης της διαδρομής δεν θα πρέπει να υπερβαίνει την χωρητικότητα του φορτηγού
- Η συνολική απόσταση που διανύει το φορτηγό της διαδρομής δεν θα πρέπει να υπερβαίνει το μέγιστο μήκος διαδρομής.
- Όλα τα φορτηγά πρέπει να ξεκινάνε και επιστρέφουν στην αποθήκη.

Ο σκοπός του προβλήματος είναι η μεγιστοποίηση του κέρδους που θα συλλεχθεί από τους πελάτες. Για το λόγο αυτό οι πελάτες που θα εισαχθούν στην διαδρομή επιλέγονται με κύριο κριτήριο το κέρδος και με δεύτερο κριτήριο την απόσταση. Παρακάτω απεικονίζεται γραφικά η διαδικασία και κάθε φορά εξετάζεται αν ο πελάτης εξυπηρετεί τους παρακάτω περιορισμούς του προβλήματος. Πρέπει να τηρούνται ταυτόχρονα οι παρακάτω συνθήκες για κάθε πελάτη.

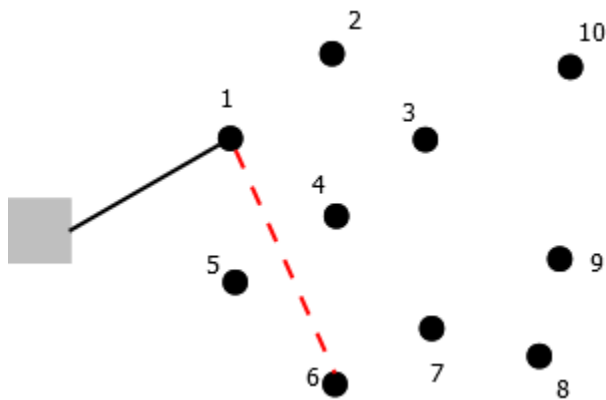
1. $Q + d(n) \leq Q_{max}$, δηλαδή η τρέχων ζήτηση της διαδρομής συν την ζήτηση του πελάτη που έχει επιλεγεί να μην υπερβαίνει την συνολική χωρητικότητα του φορτηγού.
2. $K + T(B, n) + T(n, 1) \leq T_{max}$, δηλαδή το τρέχων κόστος της διαδρομής συν την απόσταση να μεταβούμε από τον προηγούμενο πελάτη στον τωρινό συν την απόσταση από τον τωρινό πελάτη μπορούμε να γυρίσουμε πίσω στην αποθήκη να μην υπερβαίνει το συνολικό μήκος που μπορεί να διανύσει ένα φορτηγό.

Βήμα 0. Βρισκόμαστε στην αποθήκη με 2 διαθέσιμα οχήματα και επιλέγουμε τον πιο ωφέλιμο πελάτη βάση το κέρδος που θα μας προσφέρει. Έστω ότι αυτός είναι ο πελάτης 1.



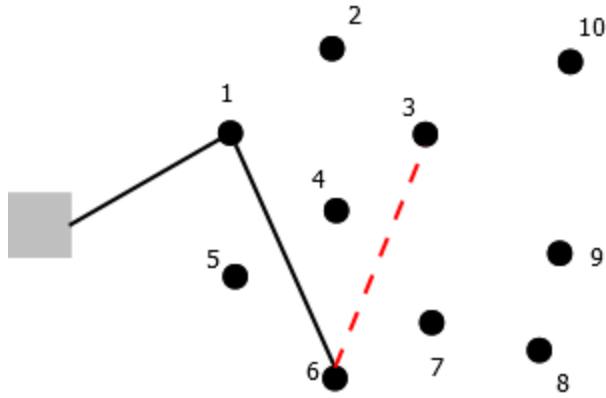
Σχήμα 4.3.3. 1

Βήμα 1. Ελέγχουμε αν ο πελάτης ικανοποιεί τους περιορισμούς χωρητικότητας και μέγιστου μήκους διαδρομής. Έστω ότι τους ικανοποιεί. Στην συνέχεια εξετάζουμε τον επόμενο πελάτη με το μεγαλύτερο κέρδος. Έστω ότι αυτός ο πελάτης είναι ο 6. Ελέγχουμε αν ικανοποιεί τους περιορισμούς.



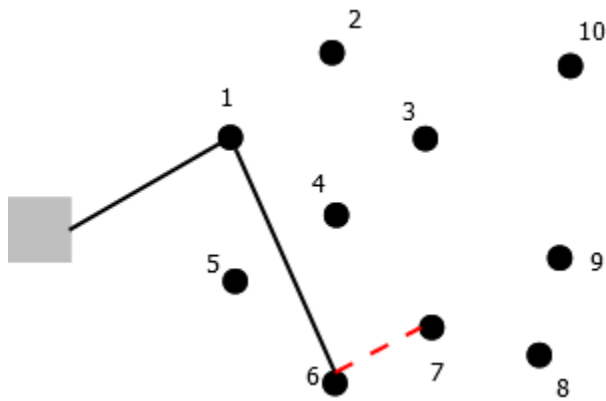
Σχήμα 4.3.3.2

Βήμα 2. Έγινε ο έλεγχος των περιορισμών και έστω ότι τους ικανοποιεί. Ο επόμενος πιο ωφέλιμος πελάτης είναι ο 3. Ελέγχουμε αν ικανοποιεί τους περιορισμούς.



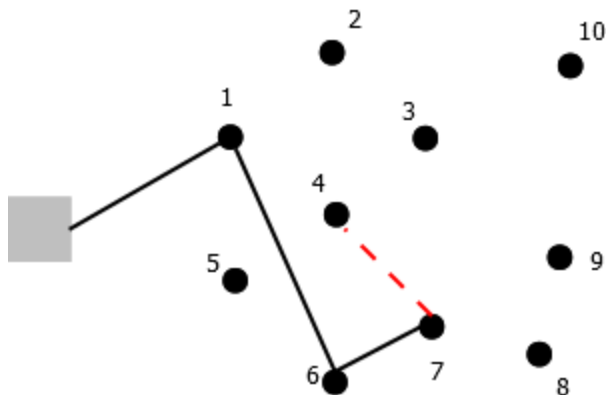
Σχήμα 4.3.3.3

Βήμα 3. Έγινε ο έλεγχος των περιορισμών και έστω ότι δεν ικανοποιείται ένας από τους περιορισμούς. Ο πελάτης που θα επιλεγεί τώρα θα γίνει με βάση το κριτήριο της απόστασης και όχι του κέρδους. Ο πιο κοντινός πελάτης στο 6 είναι ο 7. Ελέγχουμε αν ικανοποιεί τους περιορισμούς.



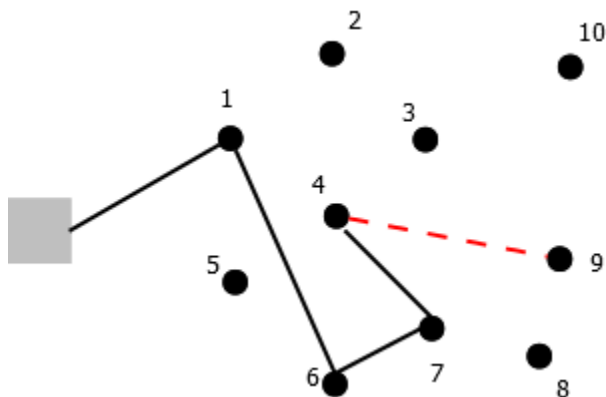
Σχήμα 4.3.3.4

Βήμα 4. Ο πελάτης 7 ικανοποιεί τους περιορισμούς. Έστω ο πελάτης με το αμέσως μεγαλύτερο κέρδος είναι ο 4. Ελέγχουμε αν ικανοποιεί τους περιορισμούς.



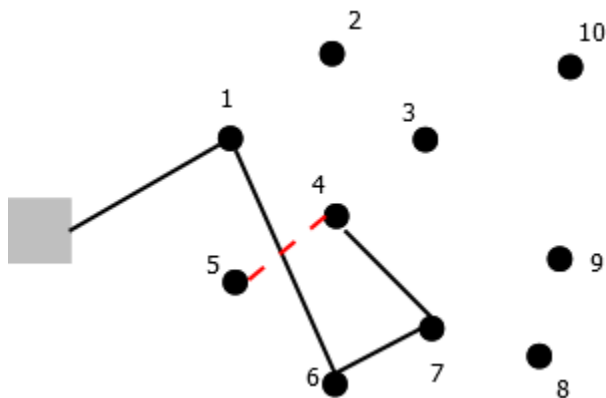
Σχήμα 4.3.3.5

Βήμα 5. Έγινε ο έλεγχος των περιορισμών και έστω ότι τους ικανοποιεί. Έστω ότι ο πιο ωφέλιμος πελάτης είναι ο 9.



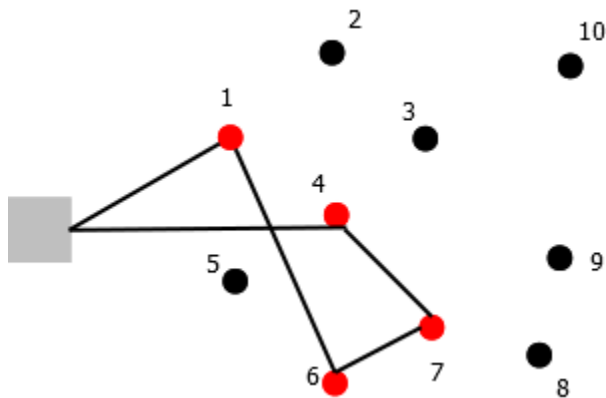
Σχήμα 4.3.3.6

Βήμα 6. Έγινε ο έλεγχος των περιορισμών και έστω ότι δεν τους ικανοποιεί. Ο πελάτης που θα επιλεγεί τώρα θα γίνει με βάση το κριτήριο της απόστασης και όχι του κέρδους. Ο πιο κοντινός πελάτης στο 4 είναι ο 5. Ελέγχουμε αν ικανοποιεί τους περιορισμούς.



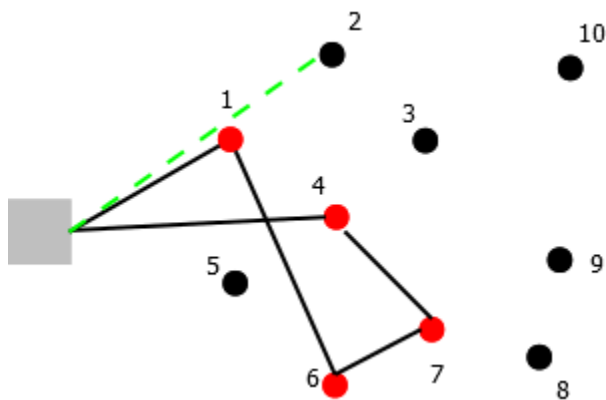
Σχήμα 4.3.3.7

Βήμα 7. Έγινε ο έλεγχος των περιορισμών και έστω ότι δεν τους ικανοποιεί. Αφού και ο πιο κοντινός πελάτης παραβιάζει τους περιορισμούς η διαδρομή τελειώνει.



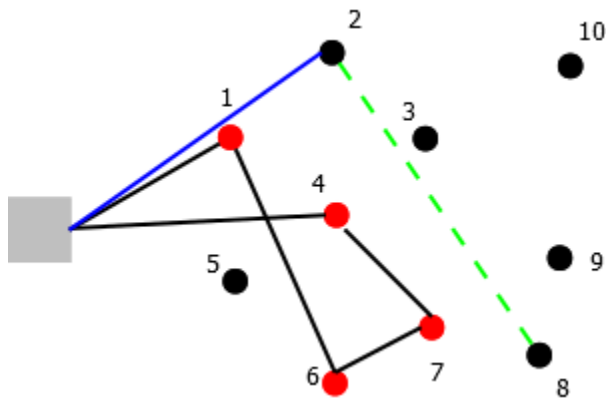
Σχήμα 4.3.3.8

Βήμα 8. Από την αποθήκη θα ξεκινήσει νέο φορτηγό. Έστω ότι ο πιο ωφέλιμος πελάτης είναι ο 2.



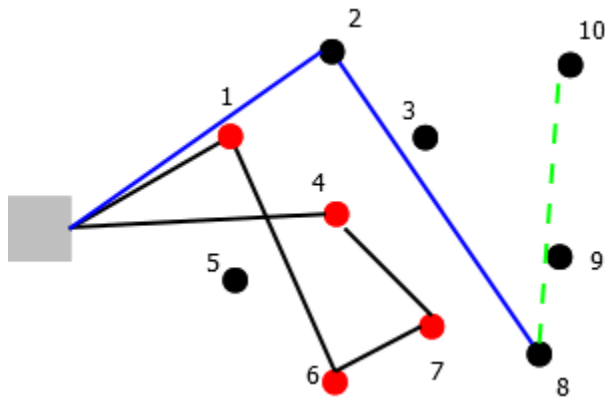
Σχήμα 4.3.3.9

Βήμα 9. Έγινε ο έλεγχος των περιορισμών και έστω ότι τους ικανοποιεί. Ο επόμενος πιο ωφέλιμος πελάτης είναι ο 8. Ελέγχουμε αν ικανοποιεί τους περιορισμούς.



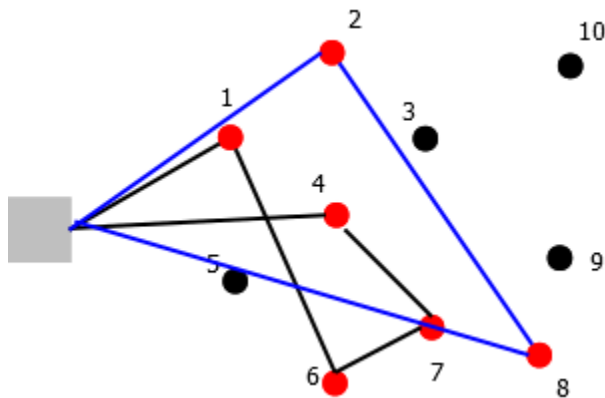
Σχήμα 4.3.3.10

Βήμα 10. Έγινε ο έλεγχος των περιορισμών και έστω ότι τους ικανοποιεί. Ο επόμενος πιο ωφέλιμος πελάτης είναι ο 10. Ελέγχουμε αν ικανοποιεί τους περιορισμούς.



Σχήμα 4.3.3.11

Βήμα 11. Έγινε ο έλεγχος των περιορισμών και έστω ότι δεν τους ικανοποιεί. Δεν υπάρχει άλλος πελάτης να ελεγχθεί συνεπώς το όχημα επιστρέφει στην αποθήκη.



Σχήμα 4.3.3.12

4.4 Εύρεση της βέλτιστης λύσης με τη χρήση του αλγορίθμου μεταβλητής γειτονιάς αναζήτησης

Η προηγούμενη μας λύση μπορεί να είναι εφικτή αλλά κατά πάσα πιθανότητα δεν είναι βέλτιστη. Για να βελτιστοποιηθεί το συνολικό κέρδος που θα συλλεχθεί από τους πελάτες γίνεται χρήση του αλγορίθμου μεταβλητής γειτονιάς αναζήτησης (Variable Neighborhood Search). Ο αλγόριθμος μεταβλητής γειτονιάς αναζήτησης αναζητά πιθανές βέλτιστες λύσεις σε όλο και μεγαλύτερες γειτονιές. Στην περίπτωση μας, θα αναζητήσουμε λύσεις που θα διαφέρουν από την τρέχουσα λύση. Η μέθοδος βασίζεται στην τοπική αναζήτηση, όπως οι

ανταλλαγές κόμβων μεταξύ της διαδρομής με τους πελάτες που δεν έχουν εισαχθεί στην διαδρομή και με την επανατοποθέτηση νέων πελατών στην διαδρομή.

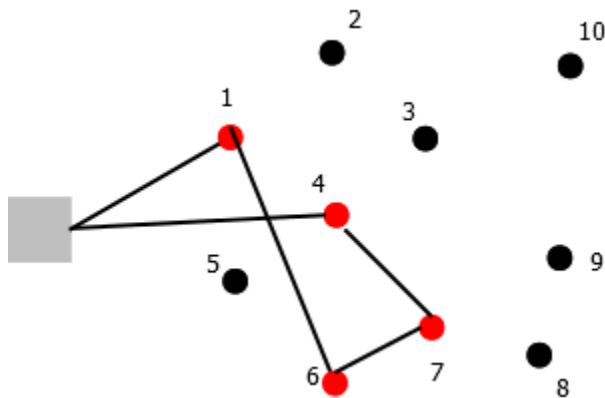
Σε πρώτη φάση χρειαζόμαστε τους πίνακες *diadromi*, *Apostasis* και *QQ* από το αρχικό κομμάτι. Στη συνέχεια ορίζουμε νέες μεταβλητές και πίνακες που θα μας χρειαστούν. Κατά την εκτέλεση του αλγορίθμου οι μετατροπές που γίνονται αποθηκεύονται σε νέους πίνακες, που δηλώνονται πριν την εκτέλεση του αλγορίθμου. Όταν βρεθεί μια καλύτερη λύση τότε ανανεώνονται οι πίνακες *diadromi*, *Apostasis* και *QQ*. Αν δεν βρεθεί κάποια καλύτερη λύση θα συνεχιστούν οι μετατροπές στους νέους πίνακες.

Ορίζουμε έναν συγκεκριμένο αριθμό επαναλήψεων όπου πραγματοποιούνται οι τοπικές αναζητήσεις που αναλύουμε παρακάτω. Αρχικά ξεκινάμε από την τοπική αναζήτηση 2-1 exchange και ολοκληρώνουμε με την 1-0 relocate.

4.4.1 Εφαρμογή τοπικής αναζήτησης 2-1 exchange (2-1 ανταλλαγή)

Στο πρώτο τμήμα του αλγορίθμου εξετάζουμε αν μπορούμε να βελτιώσουμε το συνολικό κέρδος εισάγοντας σε κάθε επανάληψη έναν επιπλέον πελάτη. Αρχικά δημιουργούμε τους βοηθητικούς πίνακες, οι οποίοι ονομάζονται *New_route*, *New_Q*, *New_apostasis* που θα αποθηκεύουν όποιες αλλαγές γίνονται. Σε κάθε γραμμή υπολογίζουμε από τον πίνακα *Apostasis* τον ακριβότερο πελάτη, δηλαδή τον πελάτη που έχει την μεγαλύτερη απόσταση. Στη συνέχεια διώχνουμε τον πελάτη με τη μεγαλύτερη απόσταση και βάζουμε μέσα στην διαδρομή μας δυο επιπλέον πελάτες, οι οποίοι δεν ανήκουν σε καμία διαδρομή.

Ας υποθέσουμε ότι η μεγαλύτερη μετάβαση είναι το τόξο 1-6. Διώχνουμε τον πελάτη 6 και ψάχνουμε δύο πελάτες για να εισαχθούν στη διαδρομή. Ο πρώτος πελάτης που θα εισαχθεί θα είναι αυτός με το μεγαλύτερο κέρδος και ο δεύτερος πελάτης θα είναι αυτός με την χαμηλότερη ζήτηση.



Σχήμα 4.4.1.1: Αρχική λύση

Diadromi(i)=

0	1	6	7	4	0
---	---	---	---	---	---

Apostasis(i)=

0	20	42	15	18	38
---	----	-----------	----	----	----

QQ(i)=

0	15	36	12	22	0
---	----	----	----	----	---

Έστω ότι ο πελάτης 3 είναι αυτός με το μεγαλύτερο κέρδος και ο πελάτης 5 με την μικρότερη ζήτηση. Γίνεται έλεγχος των περιορισμών. Υποθέτουμε ότι ικανοποιούνται οι περιορισμοί. Επομένως η διαδρομή που δημιουργείται με τα αντίστοιχα διανύσματα της απόστασης και της ζήτησης είναι:

New_route(i)=

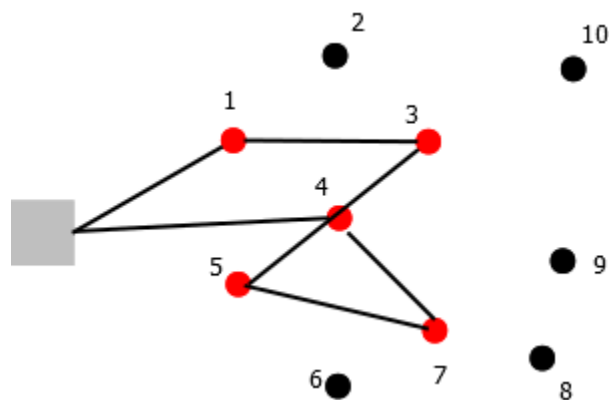
0	1	3	5	7	4	0
---	---	---	---	---	---	---

New_apostasis(i)=

0	20	32	16	15	18	38
---	----	----	----	----	----	----

New_Q(i)=

0	15	24	17	12	22	0
---	----	----	----	----	----	---



Σχήμα 4.4.1.2: Εφαρμογή τοπικής αναζήτησης 2-1 ανταλλαγή

Ελέγχουμε τους περιορισμούς της νέας διαδρομής που δημιουργείται, δηλαδή το άθροισμα όλων των αποστάσεων του διανύσματος να είναι μικρότερο του T_{\max} και το άθροισμα του διανύσματος της ζήτησης να είναι μικρότερο του Q_{\max} , και αν είναι εφικτή τότε στα διανύσματα *Diadromi*, *Apostasis* και *QQ* θα εκχωρηθούν τα διανύσματα *New_route*, *New_Apostasis* και *New_Q* αντίστοιχα. Σε αντίθετη περίπτωση, θα διατηρηθούν οι τιμές που βρήκαμε προηγουμένως.

4.4.2 Εφαρμογή τοπικής αναζήτησης 1-0 relocate (1-0 επανατοποθέτηση)

Στο δεύτερο τμήμα του αλγορίθμου εξετάζουμε εάν μπορούμε να βελτιώσουμε και άλλο το κέρδος της διαδρομής εισάγοντας σε κάθε επανάληψη έναν επιπλέον πελάτη στη διαδρομή μας. Αρχικά κρατάμε τα αποτελέσματα που βρήκαμε από το 2-1 exchange και δημιουργούμε νέους πίνακες που θα αποθηκεύουν όποιες αλλαγές γίνονται. Στην συνέχεια, επιλέγουμε τον πελάτη με το αμέσως μεγαλύτερο κέρδος και τον τοποθετούμε σε όλες τις θέσεις της διαδρομής. Παράλληλα αποθηκεύονται σε έναν πίνακα οι νέες αποστάσεις. Επιλέγουμε την μετάβαση που θα μας στοιχίσει λιγότερο και τοποθετούμε τον νέο πελάτη μας εκεί. Παρακάτω ακολουθεί ένα παράδειγμα.

Τα διανύσματα *New_route*, *New_apostasis* και *New_Q* δημιουργήθηκαν από το 2-1 ανταλλαγή.

New_route(i)=

0	1	3	5	7	4	0
---	---	---	---	---	---	---

New_apostasis(i)=

0	20	32	16	15	18	38
---	----	----	----	----	----	----

New_Q(i)=

0	15	24	17	12	22	0
---	----	----	----	----	----	---

Ας υποθέσουμε ότι ο πελάτης 8 είναι αυτός με το πιο υψηλό κέρδος. Ο πελάτης 8 θα τοποθετηθεί εκεί όπου έχει το μικρότερο κόστος. Στο παράδειγμά μας θα τοποθετηθεί δεξιά του πελάτη 7. Γίνεται έλεγχος των περιορισμών και έστω ότι τους ικανοποιεί. Επομένως, η νέα διαδρομή που δημιουργείτε με τα αντίστοιχα διανύσματα της απόστασης και της ζήτησης είναι:

New_route1(i)=

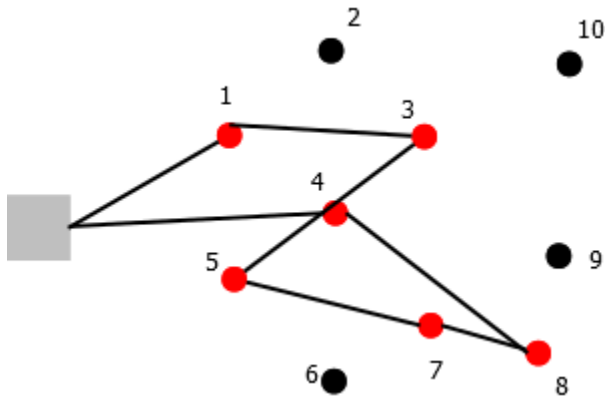
0	1	3	5	7	8	4	0
---	---	---	---	---	---	---	---

New_apostasis1(i)=

0	20	32	16	15	12	25	38
---	----	----	----	----	----	----	----

New_Q1(i)=

0	15	24	17	12	4	22	0
---	----	----	----	----	---	----	---



Σχήμα 4.4.2.1: Εφαρμογή τοπικής αναζήτησης 1-0 Επανατοποθέτηση

4.5 Τερματισμός αλγορίθμου και ψευδοκώδικας

Ο αλγόριθμος της μεταβλητής γειτονίας αναζήτησης (VNS) επαναλαμβάνεται για ένα συγκεκριμένο αριθμό επαναλήψεων, τον οποίο τον καθορίζουμε εμείς. Όσο μεγαλύτερος ο αριθμός των επαναλήψεων, τόσο αυξάνονται οι πιθανότητες να βρεθούν καλύτερες λύσεις. Έτσι κάθε βέλτιστη λύση ανά επανάληψη αποθηκεύεται και μετά το πέρας των επαναλήψεων συγκρίνονται το κόστος της διαδρομής, η ζήτηση της διαδρομής και το κέρδος της διαδρομής που έχει συλλεχθεί. Σαν βέλτιστη λύση κρατάμε αυτή που θα μας δώσει το μεγαλύτερο κέρδος και ταυτόχρονα το πιο μικρό κόστος και την πιο μικρή ζήτηση.

Η διαδικασία επίλυσης του προβλήματος παρουσιάζεται στον παρακάτω ψευδοκώδικα:

Αρχικοποίηση

Εισαγωγή δεδομένων του προβλήματος

Υπολόγισε πίνακα κόστους T

Εφαρμογή του πλησιέστερου γείτονα

Εύρεση αρχικής εφικτής υποβέλτιστη λύση

Υπολόγισε το κέρδος κάθε διαδρομής

Για έναν συγκεκριμένο αριθμό επαναλήψεων

Αποθήκευσε το διάνυσμα της διαδρομής σε ένα καινούργιο διάνυσμα

Αποθήκευσε το διάνυσμα των μεταβάσεων σε ένα καινούργιο διάνυσμα

Εύρεση της μεγαλύτερης απόστασης από τον πίνακα των μεταβάσεων

Εύρεση πελάτη με το μεγαλύτερο κέρδος

Εύρεση πελάτη με την μικρότερη ζήτηση

Εφαρμογή 2-1 exchange στο καινούργιο διάνυσμα

Υπολόγισε το νέο κόστος της διαδρομής

Υπολόγισε τη νέα ζήτηση της διαδρομής

Αν νέο κόστος μικρότερο από το αρχικό και νέα ζήτηση μικρότερη από την αρχική

Κρατάμε την νέα διαδρομή στη λύση

Αποθηκεύουμε όλες τις εφικτές λύσεις

Υπολογισμός του κόστους της λύσης

Υπολογισμός της ζήτησης της λύσης

Υπολογισμός του κέρδους της λύσης

Αλλιώς

Κρατάμε την αρχική διαδρομή

Τέλος Αν

Τέλος Για

Επιλογή λύσης με αυξημένο κέρδος και με περιθώρια βελτίωσης αυτού του κέρδους

Για έναν συγκεκριμένο αριθμό επαναλήψεων

Αποθήκευσε το διάνυσμα της διαδρομής που δημιουργήθηκε από το 2-1 ανταλλαγή σε ένα νέο διάνυσμα

Αποθήκευσε το διάνυσμα των μεταβάσεων που δημιουργήθηκε από το 2-1 ανταλλαγή σε ένα νέο διάνυσμα

Επιλογή πελάτη με μεγάλο κέρδος

Εφάρμοσε 1-0 relocate στο νέο διάνυσμα

Υπολόγισε το νέο κόστος της διαδρομής

Υπολόγισε τη νέα ζήτηση της διαδρομής

Αν νέο κόστος μικρότερο από το αρχικό και νέα ζήτηση μικρότερη από την αρχική

Κρατάμε την νέα διαδρομή στη λύση

Αποθηκεύουμε όλες τις εφικτές λύσεις

Υπολογισμός του κόστους της λύσης

Υπολογισμός της ζήτησης της λύσης

Υπολογισμός του κέρδους της λύσης

Αλλιώς

Κρατάμε την διαδρομή που δημιουργήθηκε από το 2-1 ανταλλαγή

Τέλος Αν

ΚΕΦΑΛΑΙΟ 5: Περιγραφή και αναπαράσταση αποτελεσμάτων για το πρόβλημα προσανατολισμού ομάδας με χωρητικότητα (CTOP)

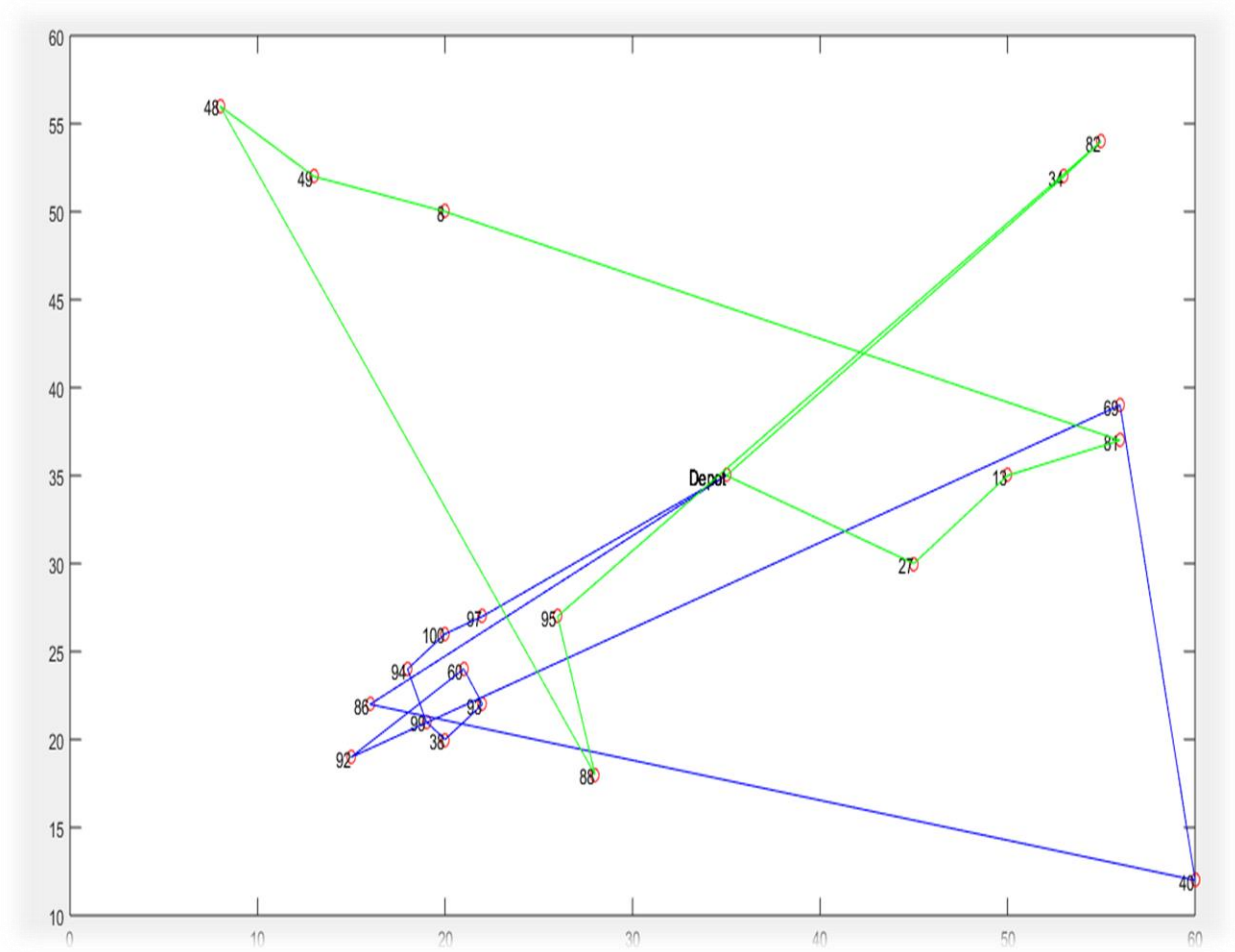
5.1 Γενική περιγραφή των αποτελεσμάτων

Με σκοπό να αξιολογήσουμε τα αποτελέσματα του αλγορίθμου τρέξαμε κάποια προβλήματα. Τα προβλήματα που διερευνήθηκαν είναι τα προβλήματα της Archetti. Κάθε πρόβλημα περιέχει από 50 έως 199 κόμβους. Τα δεδομένα των προβλημάτων ήταν ο αριθμός των πελατών, ο μέγιστος αριθμός οχημάτων, η μέγιστη απόσταση που μπορούσε να διανύσει κάθε όχημα, οι καρτεσιανές συντεταγμένες των κόμβων, η ζήτηση και το κέρδος. Τα χαρακτηριστικά των προβλημάτων παρουσιάζονται αναλυτικά στον παρακάτω πίνακα (Πίνακας 5.1).

Πίνακας 5.1 Δεδομένα προβλήματος

Προβλήματα	Πελάτες	Χωρητικότητα	Μέγιστο μήκος διαδρομής
p03	101	200	200
p06	51	160	200
p07	76	140	160
p08	101	200	230
p09	151	200	200
p10	200	200	200
p13	121	200	720
p14	101	200	1040
p15	151	200	200
p16	200	200	200

5.1.1 Αποτελέσματα του προβλήματος p03

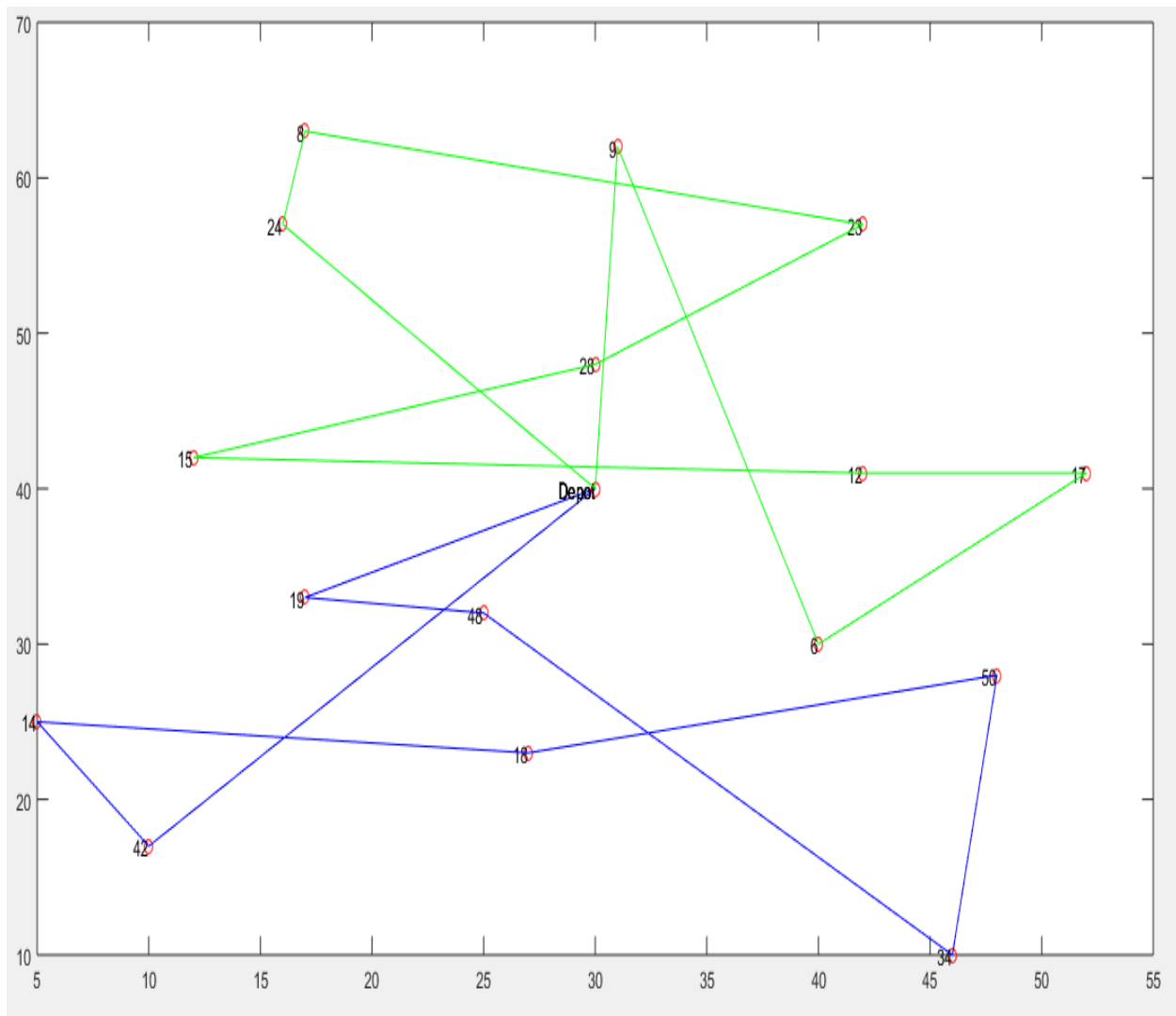


Σχήμα 5.1.1 Δρομολόγηση οχημάτων p03

Συνολικό κέρδος: 464

1	86	40	69	92	60	93	38	99	94	100	97	1
1	34	82	95	88	48	49	8	81	13	27	1	

5.1.2 Αποτελέσματα του προβλήματος p06

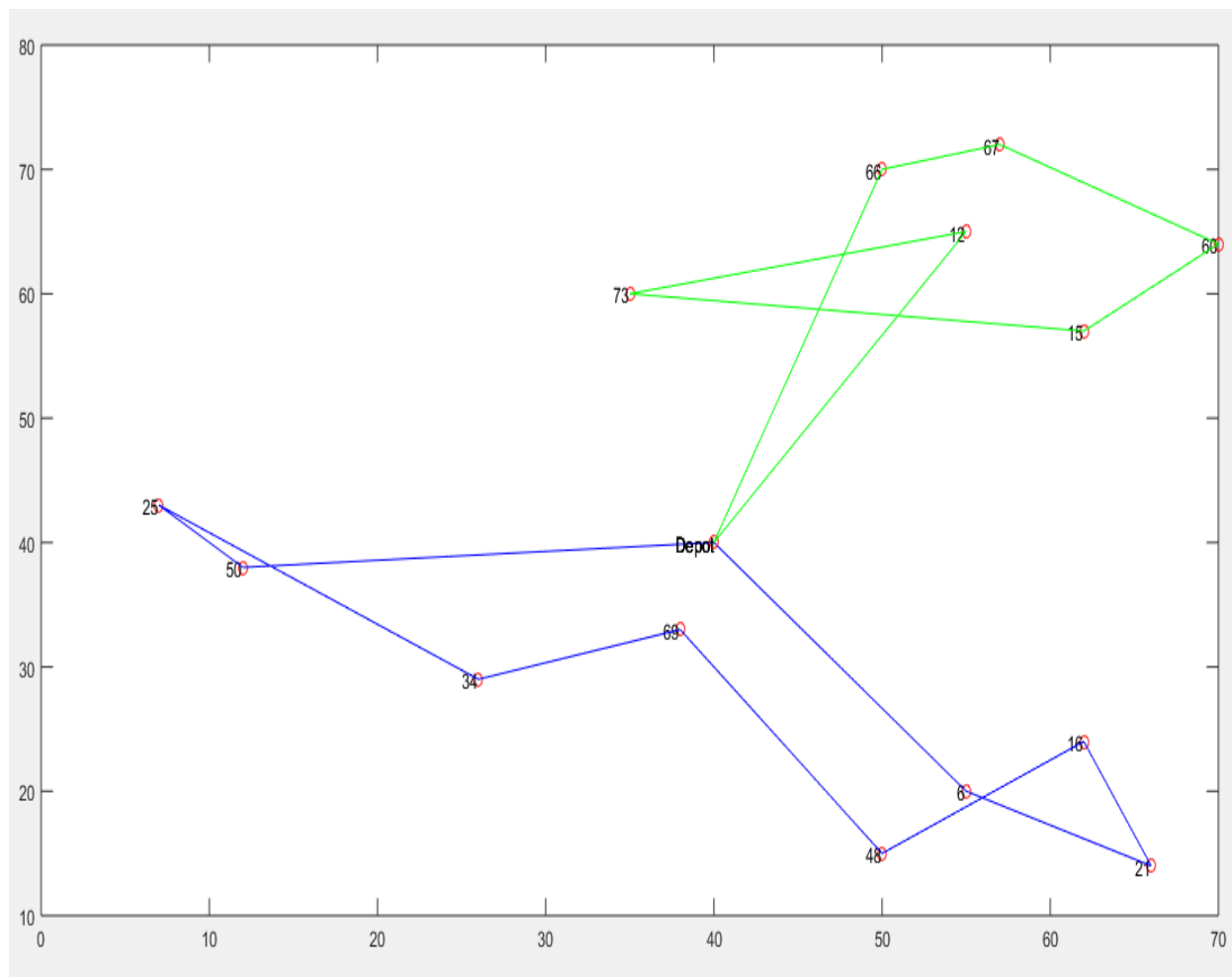


Σχήμα 5.1.2 Δρομολόγηση οχημάτων p06

Συνολικό Κέρδος: 352

1	42	14	18	50	34	48	19	1	
1	9	6	17	12	15	28	23	8	24
									1

5.1.3 Αποτελέσματα του προβλήματος p07

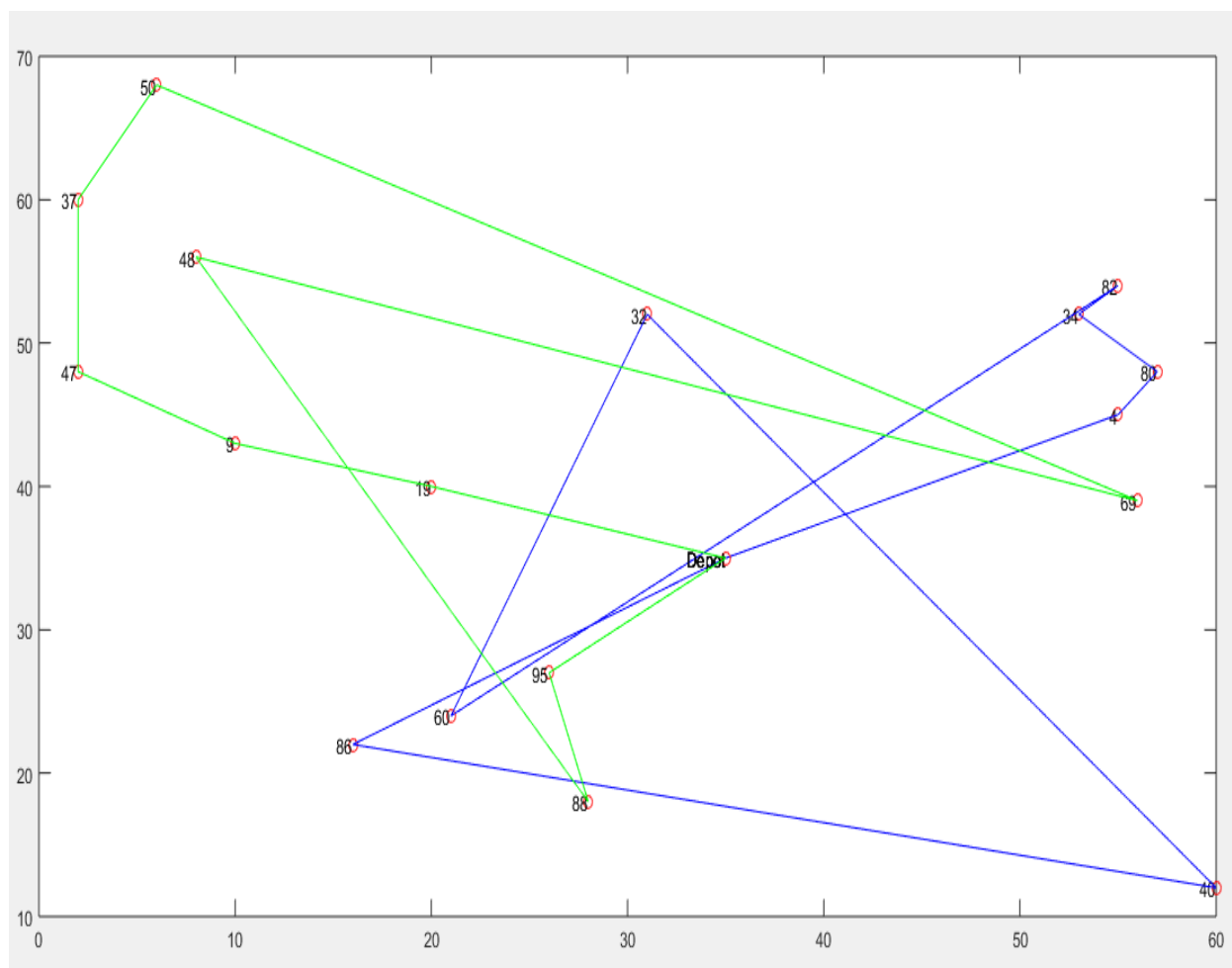


Σχήμα 5.1.3 Δρομολόγηση οχημάτων p07

Συνολικό Κέρδος: 323

1	6	21	16	48	69	34	25	50	1
1	12	73	15	60	67	66	1		

5.1.4 Αποτελέσματα του προβλήματος p08

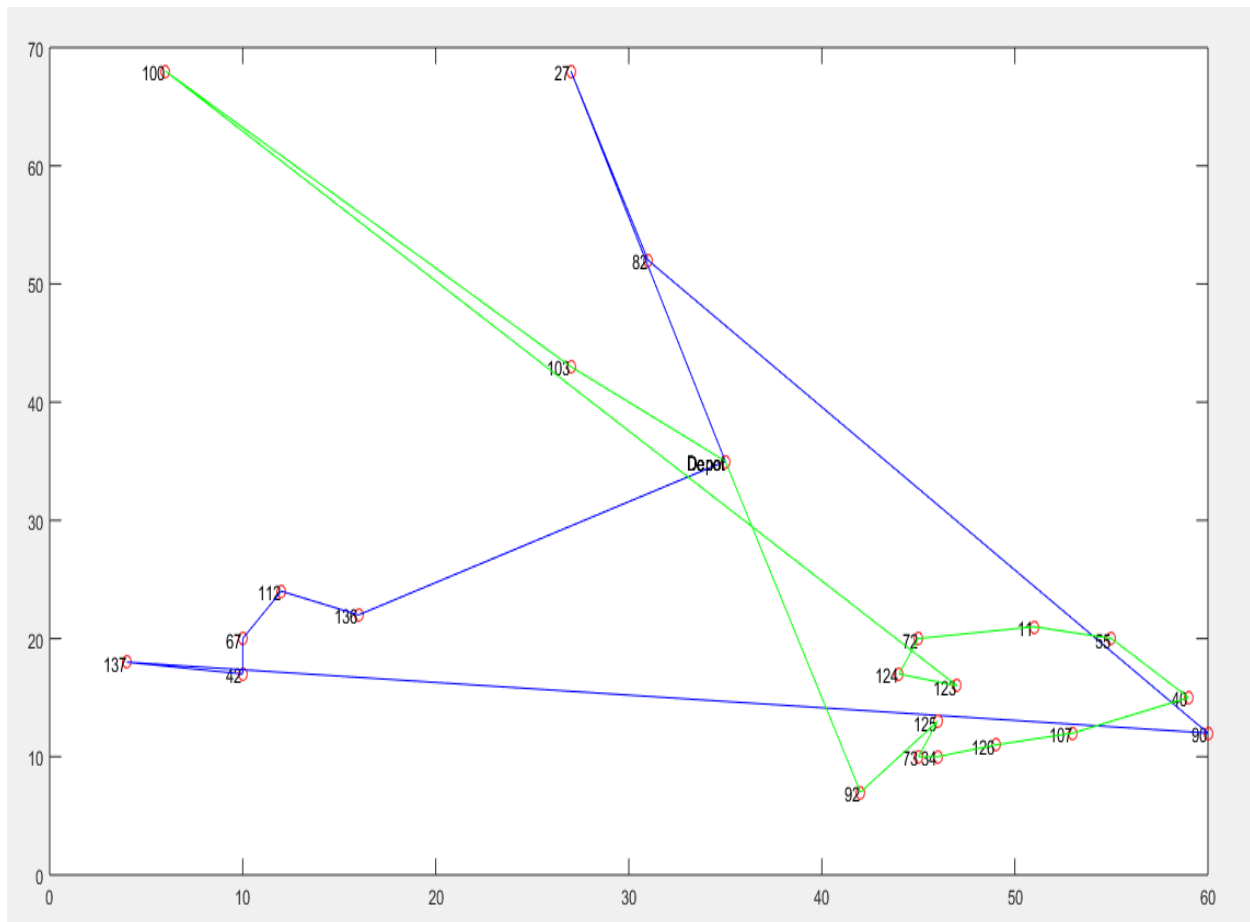


Σχήμα 5.1.4 Δρομολόγηση οχημάτων p08

Συνολικό κέρδος: 454

	1	86	40	32	60	82	34	80	4	1
1	1	95	88	48	69	50	37	47	9	19
86										
40										
32										
60										
82										
34										
80										
4										
1										

5.1.5 Αποτελέσματα του προβλήματος p09

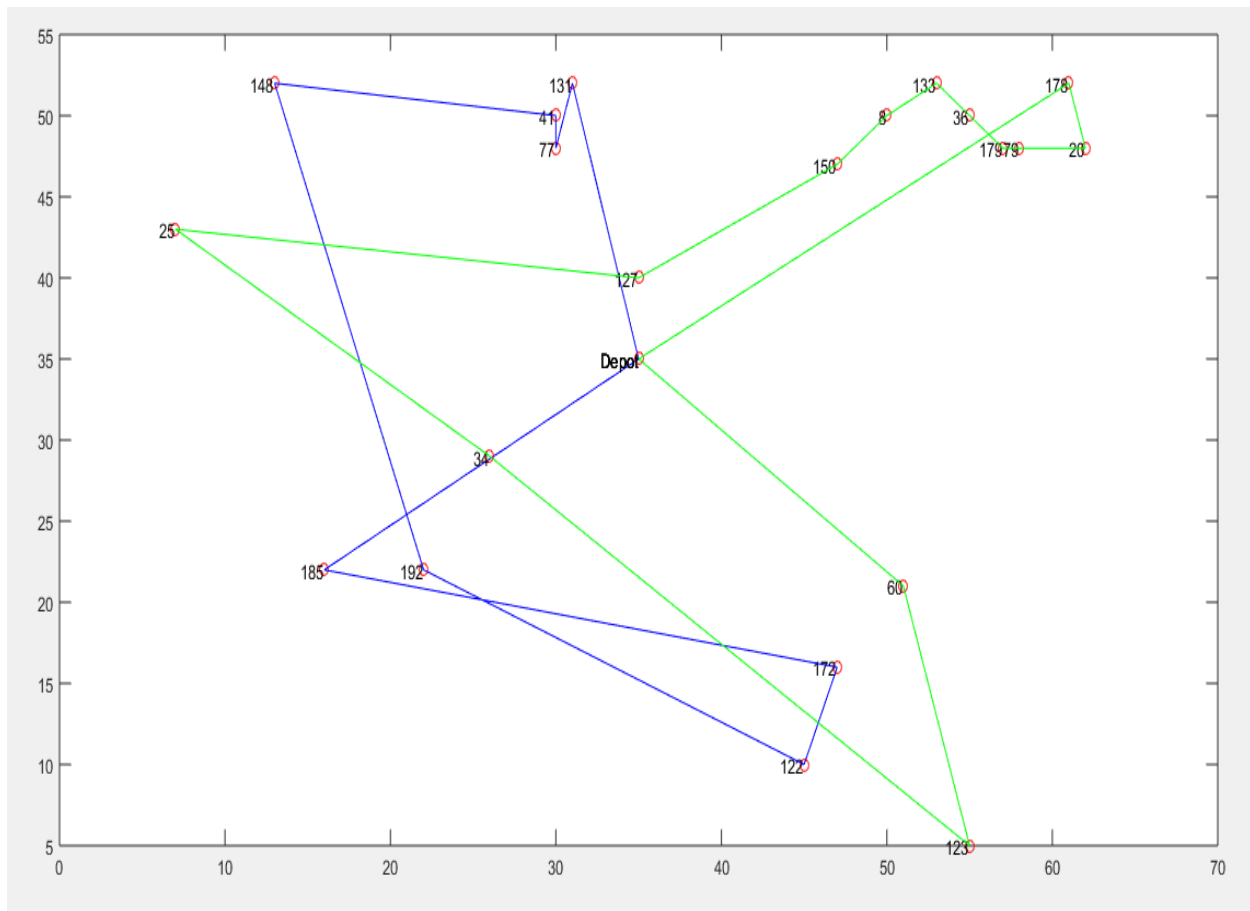


Σχήμα 5.1.5 Δρομολόγηση οχημάτων p09

Συνολικό Κέρδος: 435

1	27	82	90	137	42	67	112	136	1						
1	103	100	123	124	72	11	55	40	107	126	34	73	125	92	1

5.1.6 Αποτελέσματα του προβλήματος p10

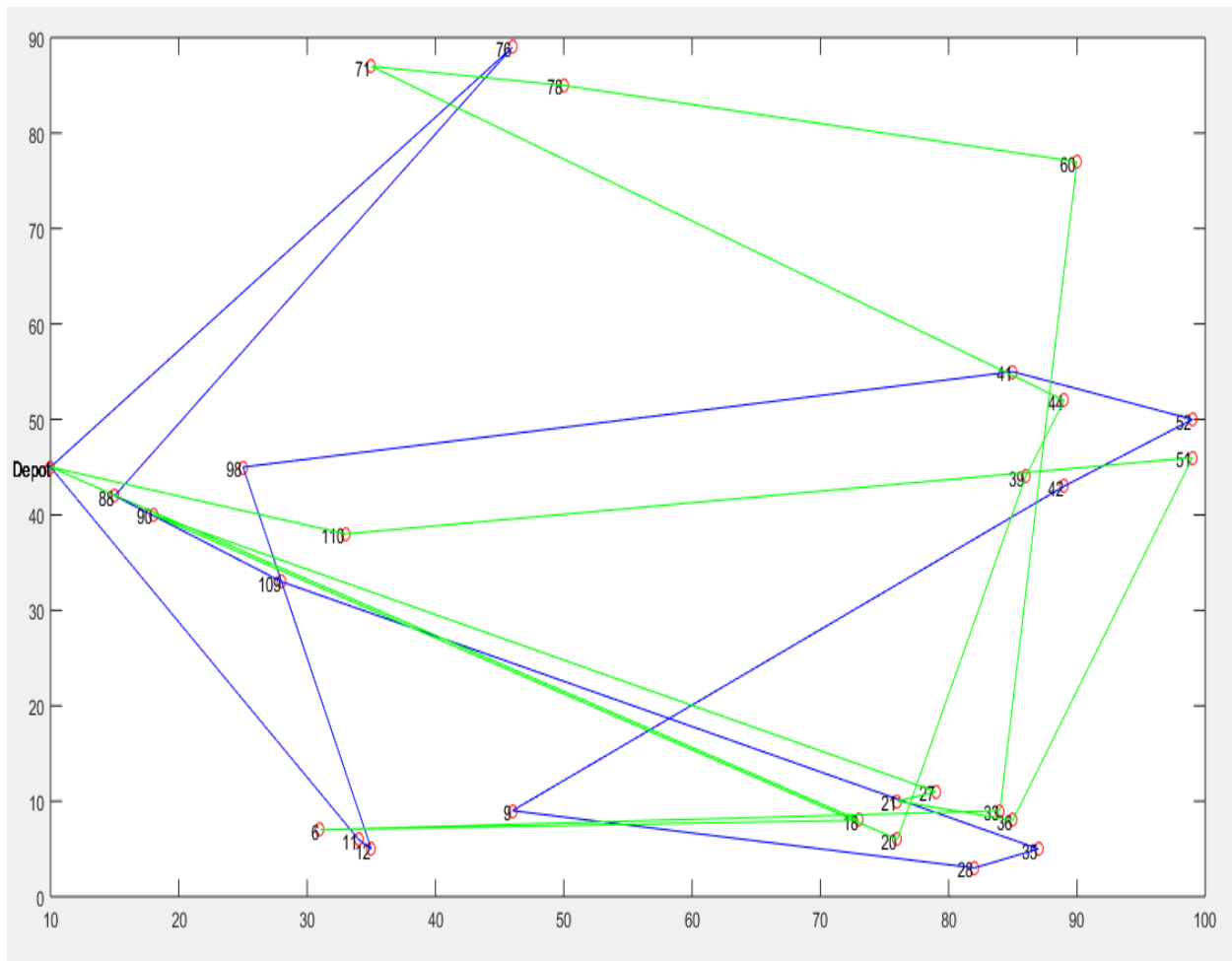


Σχήμα 5.1.6 Δρομολόγηση οχημάτων p10

Συνολικό κέρδος: 496

1	185	172	122	192	148	41	77	131	1						
1	60	123	34	25	127	150	8	133	36	179	79	20	178	1	

5.1.7 Αποτελέσματα του προβλήματος p13

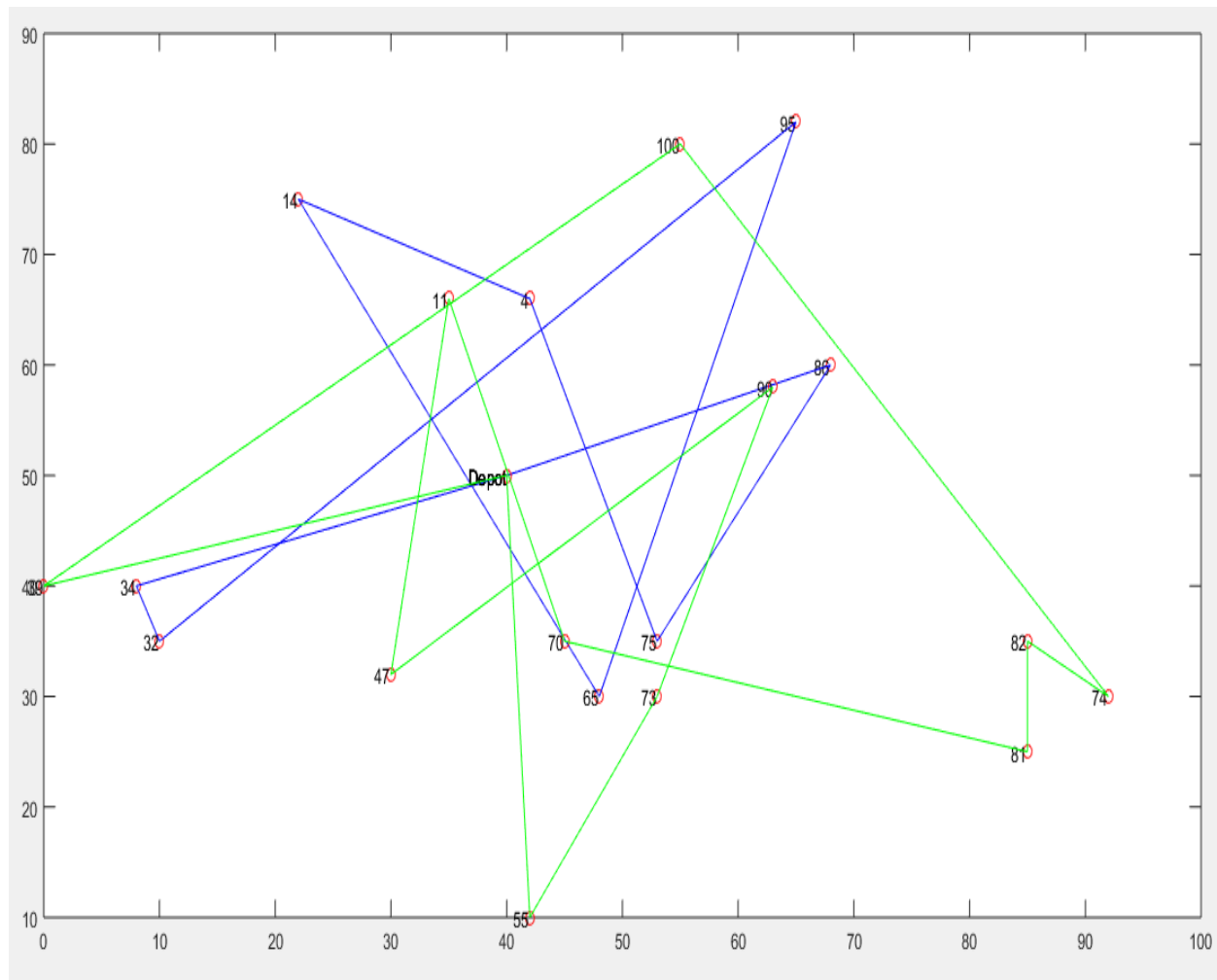


Σχήμα 5.1.7 Δρομολόγηση οχημάτων p13

Συνολικό Κέρδος: 450

1	76	88	109	35	28	9	42	52	41	98	12	11	1			
1	110	51	36	21	27	90	20	39	44	71	78	60	33	6	18	1

5.1.8 Αποτελέσματα του προβλήματος p14

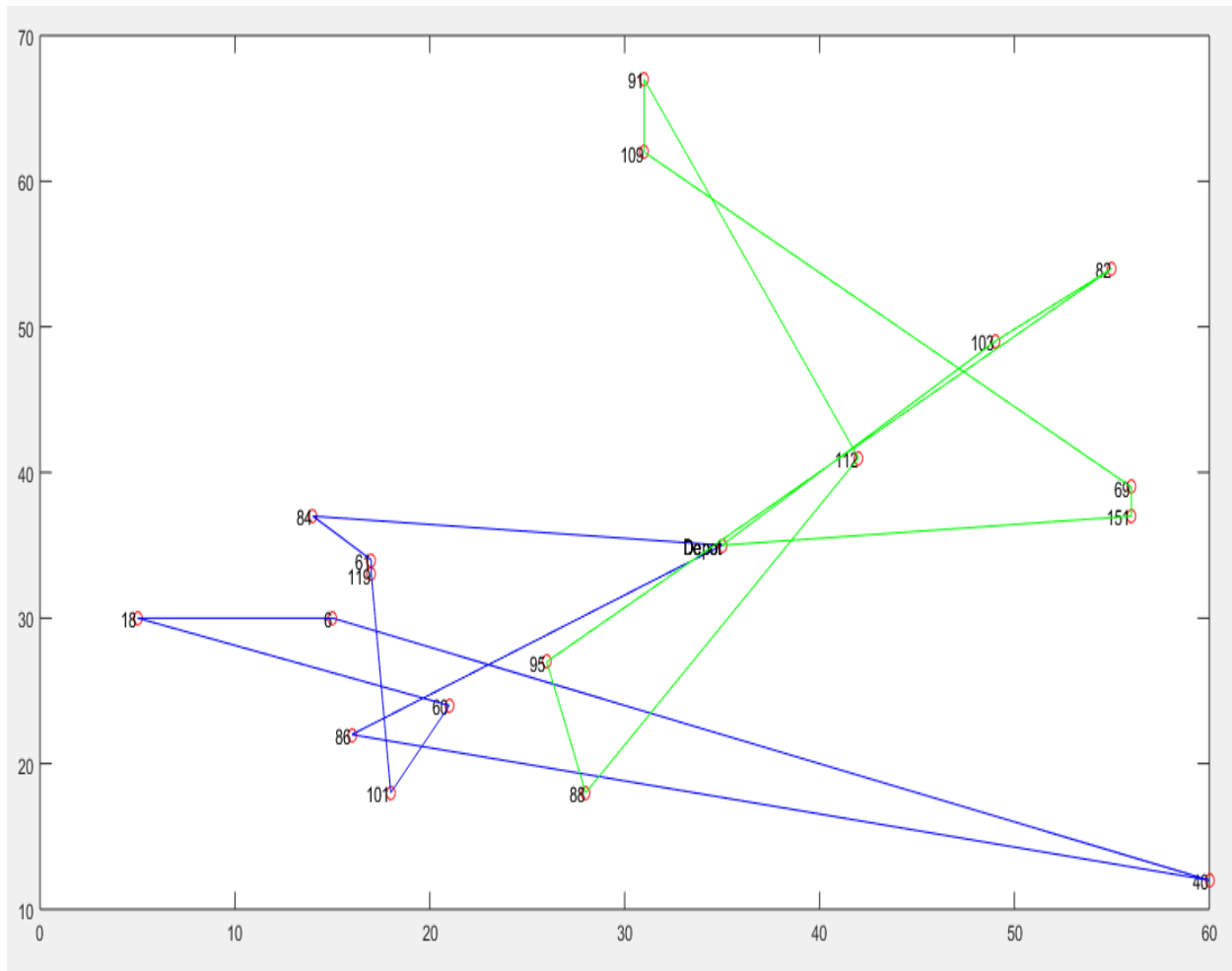


Σχήμα 5.1.8 Δρομολόγηση οχημάτων p14

Συνολικό Κέρδος: 442

1	34	32	95	65	14	4	75	86	1			
1	39	100	74	82	81	70	11	47	90	73	55	1

5.1.9 Αποτελέσματα του προβλήματος p15

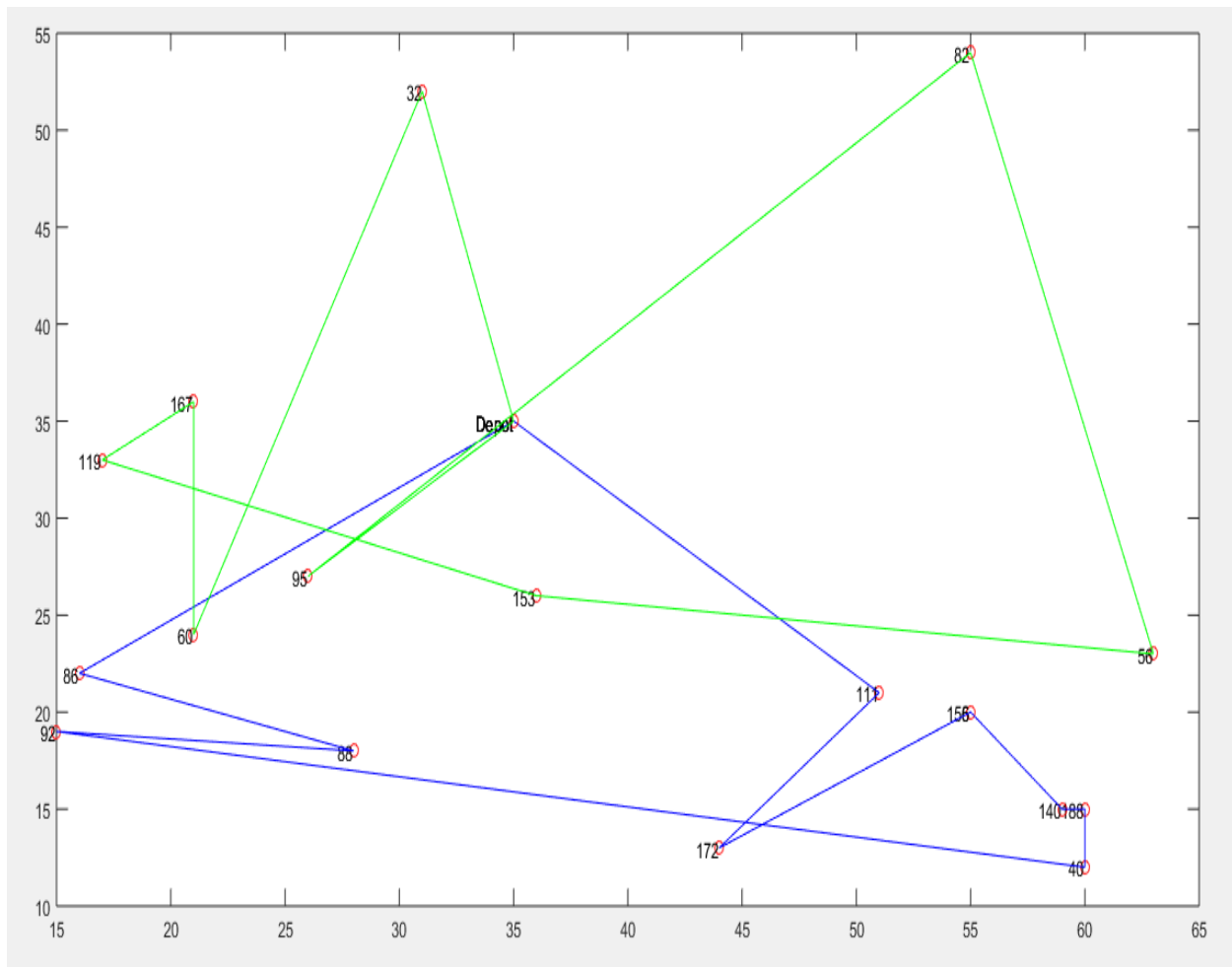


Σχήμα 5.1.9 Δρομολόγηση οχημάτων p15

Συνολικό Κέρδος: 492

1	86	40	6	18	60	101	119	61	84	1
1	103	82	95	88	112	91	109	69	151	1

5.1.10 Αποτελέσματα του προβλήματος p16



Σχήμα 5.1.10 Δρομολόγηση οχημάτων p16

Συνολικό κέρδος: 502

1	86	88	92	40	188	140	5	156	172	111	1
1	32	60	167	119	153	56	82	95	1		

5.2 Πίνακας αποτελεσμάτων

Παραπάνω παρουσιάστηκαν τα γραφήματα και οι πίνακες διαδρομών από ένα δείγμα προβλημάτων. Στη συνέχεια συνοψίζονται όλα τα αποτελέσματα στον παρακάτω πίνακα. Ο πίνακας (Πίνακας 5.2) περιλαμβάνει την αρχική εφικτή λύση, τη βέλτιστη λύση που βρέθηκε από τον αλγόριθμο Μεταβλητής Γειτονίας Αναζήτησης και η παγκοσμίως βέλτιστη λύση.

Πίνακας 5.2

Προβλήματα	Αποτελέσματα Αλγορίθμου Πλησιέστερου Γείτονα	Οχήματα στον Αλγόριθμο του Πλησιέστερου Γείτονα	Αποτελέσματα στον αλγόριθμο Μεταβλητής Γειτονίας Αναζήτησης	Οχήματα στον αλγόριθμο Μεταβλητής Γειτονίας Αναζήτησης	Παγκοσμίως βέλτιστες λύσεις	Παγκοσμίως βέλτιστα οχήματα
p03	450	2	464	2	536	2
p06	326	2	352	2	403	2
p07	314	2	323	2	377	2
p08	445	2	454	2	536	2
p09	417	2	435	2	547	2
p10	483	2	496	2	556	2
p13	433	2	450	2	513	2
p14	440	2	442	2	534	2
p15	438	2	492	2	550	2
p16	450	2	502	2	558	2

ΚΕΦΑΛΑΙΟ 6: Σύγκριση αποτελεσμάτων και συμπεράσματα

6.1 Σύγκριση αποτελεσμάτων

Στο προηγούμενο κεφάλαιο παρουσιάστηκαν τα αποτελέσματα των αλγορίθμων πλησιέστερου γείτονα και μεταβλητής γειτονίας αναζήτησης. Σε αυτό το κεφάλαιο θα γίνει αξιολόγηση αυτών των αποτελεσμάτων. Αρχικά θα παρουσιαστεί ένας πίνακας (Πίνακας 6.1) που θα συγκρίνει τα αποτελέσματα που βρήκαμε με τα παγκοσμίως βέλτιστα αποτελέσματα. Η σύγκριση των αποτελεσμάτων αναγράφεται σε ποσοστιαία απόκλιση.

Πίνακας 6.1

Προβλήματα	Πελάτες	Χωρητικότητα	Μέγιστο μήκος διαδρομής	Αποτελέσματα αλγορίθμου VNS	Παγκοσμίως βέλτιστες λύσεις	Απόκλιση (%)
p03	101	200	200	464	536	13.43
p06	51	160	200	352	403	12.65
p07	76	140	160	323	377	14.32
p08	101	200	230	454	536	15.29
p09	151	200	200	435	547	20.47
p10	200	200	200	496	556	10.79
p13	121	200	720	450	513	12.28
p14	101	200	1040	442	534	17.22
p15	151	200	200	492	550	10.54
p16	200	200	200	502	558	10.03

6.2 Συμπεράσματα

Από τον Πίνακα 6.1 οδηγούμαστε σε κάποιες στατιστικές παρατηρήσεις, βάση των οποίων μπορεί να εκτιμηθεί η ποιότητα των αποτελεσμάτων και εν συνέχεια των αλγορίθμων. Οι κυριότερες στατιστικές παρατηρήσεις είναι:

- Ο μέσος όρος απόκλισης από τη παγκοσμίως βέλτιστη λύση είναι 13.7%. Το εύρος της απόκλισης κυμαίνεται από 10.03% έως 20.47%
- Μέσω του αλγορίθμου μεταβλητής γειτονίας αναζήτησης βελτιώσαμε το κέρδος. Το συνολικό κέρδος βελτιώθηκε κατά 5.1% από το αρχικό συνολικό κέρδος.

Ο αλγόριθμος βελτιστοποίησης με τη χρήση του αλγορίθμου μεταβλητής γειτονίας αναζήτησης είναι αρκετά αποτελεσματικός. Η απόκλιση στα παραδείγματα είναι σχετικά μικρή. Η λύση με τον αλγόριθμο μεταβλητής γειτονίας αναζήτησης βελτίωσε τις αρχικές λύσεις κατά ένα ικανοποιητικό ποσοστό.

Συνοπτικά, ο κώδικας ήταν ικανοποιητικός. Είναι σίγουρο όμως ότι με τις κατάλληλες αλλαγές στον κώδικα θα έχουμε βελτίωση των αποτελεσμάτων. Μερικές διορθωτικές αλλαγές που μπορούν να γίνουν είναι:

- Να γίνει χρήση επιπλέον αλγορίθμων τοπικής αναζήτησης σε συνδυασμό με αυτούς που χρησιμοποιήσαμε.
- Να αυξηθούν οι επαναλήψεις στους αλγορίθμους τοπικής αναζήτησης.

Βιβλιογραφία

- [1] Zhixing Luo, B. C. (2013, January 2). An adaptive ejection pool with toggle-rule diversification approach for the capacitated team orienteering problem. *European Journal of Operational Research*, σσ. 673-682.
- [2] Ιωάννης Μαρινάκης, Α. Μ. (2008). *Σχεδιασμός και βελτιστοποίηση της εφοδιαστικής αλυσίδας*. Θεσσαλονίκη: Εκδόσεις σοφία.
- [3] <https://www.logistics.org.gr/4/27/136/>
- [4] Moukrim, A. B.-S.-H. (n.d.). An adaptive heuristic for Capacitated Team Orienteering Problem. σσ. 1662-1666.
- [5] Dragana Drenovac, R. R. (n.d.). The Capacited Team Orienteering Problem: Bee Colony Optimization. σσ. 43-48.
- [6] Pierre Hansen, N. M. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, σσ. 449-467.
- [7] Pierre Hansen, N. M. (2002). *Variable Neighborhood Search*. Kluwer Academic Publishers.
- [8] Thomas A. FEO, M. G. (1995). Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization* , σσ. 109-133.
- [9] Kirkpatrick, S. (1983, November 15). Optimization by Simulated Annealing: Quantitative Studies. σσ. 975-986.
- [10] Glover, F. (1989). Tabu search - Part I . *ORSA Journal on Computing* , σσ. 190-206.
- [11] Glover, F. (1990). Tabu Search- Part II. *ORSA Journal on Computing*, σσ. 4-32.