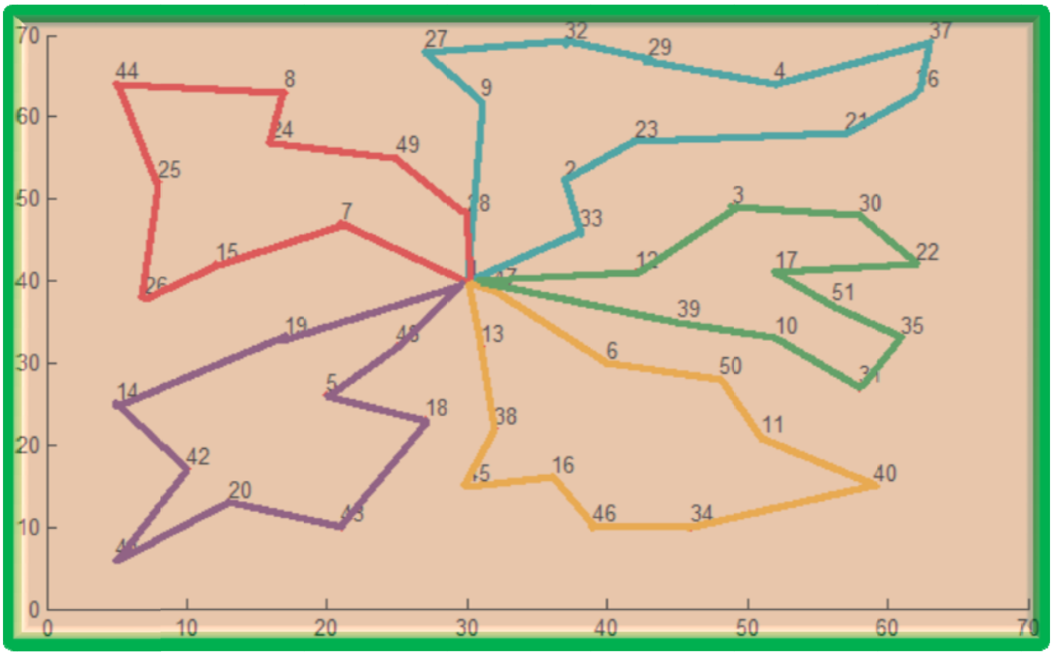
ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ρογδάκης Ιωάννης

Μεμετικός Αλγόριθμος Πολλαπλών Πληθυσμών-Νησιών στο πρόβλημα VRP



Επιβλέπων: Μαρινάκης Ιωάννης

ΧΑΝΙΑ 2009

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Γενικά οι μεταφορές σχετίζονται σχεδόν με όλα τα στάδια μιας παραγωγικής διαδικασίας αλλά και των συστημάτων διανομής των προϊόντων. Αναμφίβολα αντιπροσωπεύουν ένα σημαντικό ποσοστό του τελικού κόστους των προϊόντων. Γι' αυτό η επιστήμη της Επιχειρησιακής έρευνας ασχολείται όλο και περισσότερο με θέματα και μοντέλα βελτιστοποίησης, αφού η χρήση τους έχει αποδειχθεί καθοριστική στη διαχείριση δικτύων διανομής και γενικότερα σε όλα τα επίπεδα λειτουργίας μιας επιχείρησης. Η χρήση τέτοιων μοντέλων μπορεί να μειώσει σημαντικά τα κόστη μεταφοράς των επιχειρήσεων βοηθώντας στην εύρυθμη λειτουργία των συγκεκριμένων επιχειρήσεων αλλά και της παγκόσμιας οικονομίας γενικότερα.

Στην αύξηση της χρήσης των παραπάνω μοντέλων βοήθησε η δημιουργία λογισμικών πακέτων και πληροφοριακών συστημάτων με δυνατότητες επίλυσης πολύπλοκων προβλημάτων όμοιων με αυτών που αντιμετωπίζει μια σύγχρονη μονάδα παραγωγής. Γι' αυτό και η χρησιμοποίησή τους στις σύγχρονες βιομηχανίες διαρκώς επεκτείνεται καθιστώντας το απαραίτητο και μοναδικό εργαλείο για τον σχεδιασμό και την υλοποίηση μιας παραγωγικής διαδικασίας.

Ένας ακόμη παράγοντας επιτυχίας των μοντέλων βελτιστοποίησης είναι η διαρκής ανάπτυξη αλγορίθμων οι οποίοι καθιστούν ολοένα και πιο δυνατή την επίλυση ακόμη και των πιο δύσκολων προβλημάτων διανομής αφού όσο πιο πολλοί παράμετροι παρουσιάζονται στα προς βελτιστοποίηση προβλήματα, τόσο δυσκολότερη είναι η επίλυσή τους. Ακόμη και στα δυσκολότερα προβλήματα τα υπάρχοντα πληροφοριακά συστήματα δίνουν ικανοποιητικές λύσεις με χρόνους υπολογισμού αντίστοιχα αρκετά ικανοποιητικούς. [3]

Στην συγκεκριμένη εργασία θα εξετάσουμε μόνο προβλήματα σχετικά με τη διανομή προϊόντων μεταξύ του χώρου αποθήκευσης και των τελικών πελατών. Τα προβλήματα αυτά είναι γνωστά ως προβλήματα δρομολόγησης οχημάτων (Vehicle Routing Problems V.R.P.). Στη συνέχεια της εργασία, για λόγους συντομίας, θα χρησιμοποιείται ο όρος VRP ως συντομογραφία για τα Προβλήματα Δρομολόγησης Οχημάτων.

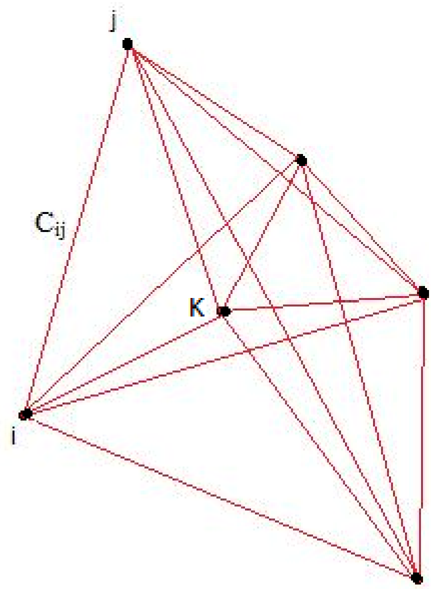
ΚΕΦΑΛΑΙΟ 2
Το Πρόβλημα Δρομολόγησης Οχημάτων (VRP)

2.1 Περιγραφή του Προβλήματος Δρομολόγησης Οχημάτων (VRP)

Ο ορισμός Vehicle Routing Problem (V.R.P.) συμπεριλαμβάνει έναν μεγάλο αριθμό προβλημάτων που σχετίζονται με τις διαδρομές που πρέπει να ακολουθήσει ένα σύνολο οχημάτων από μία αποθήκη προς έναν αριθμό πελατών. Ο στόχος που επιτυγχάνεται με την επίλυση του V.R.P είναι η ελαχιστοποίηση του κόστους μεταφοράς των προϊόντων με ταυτόχρονη ικανοποίηση της ζήτησης των πελατών, βελτιστοποιώντας τις διαδρομές ενός στόλου οχημάτων που ξεκινούν και καταλήγουν σε μία αποθήκη.

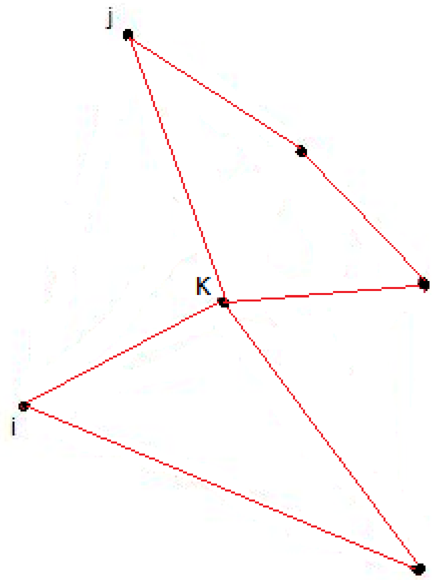
Η επίλυση, όμως, τέτοιων προβλημάτων (V.R.P.) απαιτεί τον προσδιορισμό ενός συγκεκριμένου αριθμού δρομολογίων, κάθε ένα από τα οποία θα εκτελείται αποκλειστικά από ένα όχημα το οποίο σαν έναρξη και σαν λήξη διαδρομής του θα έχει το ίδιο σημείο (αποθήκη). Το κάθε όχημα θα πρέπει να επισκέπτεται έναν πελάτη μόνο μια φορά παραδίδοντας προϊόντα σε αυτόν. Η όλη διαδικασία επίλυσης λειτουργεί προς ελαχιστοποίηση του συνολικού κόστους μεταφοράς. Κατά την επίλυση λαμβάνονται υπόψη περιορισμοί οι οποίοι περιγράφουν τις αποστάσεις μεταξύ των πελατών, τη συνολική διανυόμενη απόσταση για κάθε ένα όχημα, την χωρητικότητα των οχημάτων καθώς επίσης οτιδήποτε περιγράφει τα χαρακτηριστικά του οδικού δικτύου διανομής.

Το οδικό δίκτυο που χρησιμοποιείται για τη μεταφορά των αγαθών περιγράφεται μέσω ενός γραφήματος, τα τόξα του οποίου αντιπροσωπεύουν τις οδικές προσβάσεις ανάμεσα στους κόμβους που αντιπροσωπεύουν την αποθήκη και τους πελάτες. Κάθε τόξο χαρακτηρίζεται από το κόστος διέλευσης του (C_{ij}). Για παράδειγμα στο σχήμα 1.1 έχουμε ένα γράφημα 6 κόμβων (την αποθήκη κ και τους υπόλοιπους 5 πελάτες) με όλες τις δυνατές διαδρομές από κόμβο σε κόμβο.



Σχήμα 2.1 Όλες οι δυνατές διαδρομές

Στο επόμενο σχήμα 2.2 βλέπουμε την διαδρομή που θα ακολουθηθεί από κάθε όχημα μετά την επίλυση του προηγούμενου VRP προβλήματος. Η διαδρομή κάθε οχήματος ξεκινάει και τελειώνει στην αποθήκη κ. Δεν υπάρχει περιορισμός για τον αριθμό των τοποθεσιών που πρέπει να επισκεφτεί κάθε όχημα, εκτός του ότι πρέπει να επισκεφτεί τουλάχιστον μια τοποθεσία.



Σχήμα 2.2 Αναπαράσταση πιθανής λύσης προβλήματος VRP

2.2 Χαρακτηριστικά ενός προβλήματος VRP

Σε αυτό το σημείο είναι απαραίτητο να γίνει περιγραφή των κύριων χαρακτηριστικών ενός προβλήματος VRP (πελάτες, αποθήκη, οχήματα αλλά και οδηγοί), και τις διάφορες μεταβλητές που μπορούν να εισαχθούν στο πρόβλημα καθώς και τους πιθανούς στόχους που θέλουμε να επιτύχουμε κατά τη βελτιστοποίηση του προβλήματος το οποίο αντιμετωπίζουμε.

Οχήματα

Η μεταφορά και διακίνηση προϊόντων εκτελείται με την χρησιμοποίηση ενός στόλου οχημάτων, του οποίου η σύνθεση μπορεί να καθοριστεί σύμφωνα με τις απαιτήσεις των πελατών. Κάθε όχημα έχει συγκεκριμένη χωρητικότητα εκφρασμένη σαν μέγιστο βάρος, ή όγκο, ή αριθμό παλετών ή προϊόντων που μπορεί να μεταφέρει το κάθε όχημα. Το κόστος που συνδέεται με την χρήση του οχήματος εκφράζεται π.χ. σαν κόστος /μονάδα απόστασης κλπ.

Πελάτες

Οι κορυφές του γραφήματος (εκτός αυτής της αποθήκης) υποδεικνύουν τις θέσεις των πελατών. Ο κάθε πελάτης έχει συγκεκριμένη γνωστή ζήτηση η οποία πρέπει να ικανοποιηθεί εξ’ ολοκλήρου.

Δρομολόγια

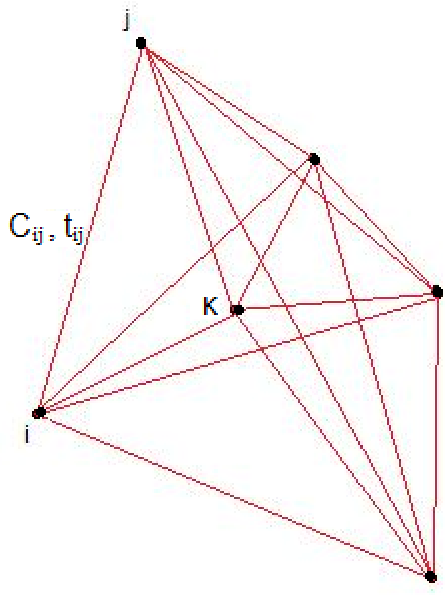
Τα δρομολόγια που εκτελούνται για την εξυπηρέτηση των πελατών ξεκινούν και καταλήγουν σε μια αποθήκη. Τα δρομολόγια πρέπει να ικανοποιούν λειτουργικούς περιορισμούς που εξαρτώνται από την ποσότητα των μεταφερόμενων προϊόντων, και τα χαρακτηριστικά των οχημάτων. Για παράδειγμα: Σε κάθε διαδρομή η μεταφερόμενη ποσότητα δεν θα πρέπει να ξεπερνά την χωρητικότητα του οχήματος.

Οδηγοί

Κατά την μοντελοποίηση, το σύνολο των περιορισμών που αναφέρεται στα οχήματα μπορεί να περικλείει και περιορισμούς που προκύπτουν από τους οδηγούς. Οι οδηγοί των οχημάτων είναι απαραίτητο να ικανοποιούν διάφορους περιορισμούς που έχουν να κάνουν με συμβόλαια εργατικών ενώσεων καθώς και κανονισμούς της εταιρείας για την οποία εργάζονται. Ο χρόνος εργασίας κατά τη διάρκεια της ημέρας, ο αριθμός και η διάρκεια των διαλειμμάτων και οι υπερωρίες αποτελούν ορισμένους μόνο από τους "κανόνες" τους οποίους πρέπει να τηρούν οι οδηγοί των οχημάτων. [3]

Η εκτίμηση του συνολικού κόστους της κάθε διαδρομής και ο έλεγχος των λειτουργικών περιορισμών που επιβάλλονται σε αυτές, απαιτεί την γνώση του κόστους μετάβασης από τον ένα πελάτη στον άλλον και το κόστος διαδρομής από και προς την αποθήκη.

Παρατηρούμε λοιπόν στο γράφημα-που αναπαριστά το δίκτυο διανομής, τις κορυφές-κόμβους του με τους πελάτες και την αποθήκη και τα τόξα με τις διαδρομές, που μπορούν να υφίστανται ανάμεσα σε αποθήκη και πελάτες, το κόστος μετάβασης από έναν κόμβο i σε έναν κόμβο j, είναι το c_{ij} το οποίο ανήκει στη συντομότερη διαδρομή μεταξύ i και j. Ο δε χρόνος μετάβασης t_{ij} , ισούται με το άθροισμα των χρόνων μετάβασης όλων των τόξων που ανήκουν στην συντομότερη διαδρομή μεταξύ i και j. π.χ. σχήμα 2.3



Σχήμα 2.3

2.3 Στόχοι ενός προβλήματος VRP

Για το V.R.P μπορούν να τεθούν διάφοροι στόχοι, οι βασικότεροι είναι οι ακόλουθοι:

- Η ελαχιστοποίηση του συνολικού κόστους μεταφοράς, το οποίο εξαρτάται από την συνολική απόσταση ή τον συνολικό χρόνο μεταφοράς.

- Το φορτίο που μεταφέρεται σε οποιαδήποτε διαδρομή να μην ξεπερνά την χωρητικότητα του αντίστοιχου οχήματος.

- Ο συνολικός χρόνος ενός δρομολογίου να μην υπερβαίνει το όριο που έχει τεθεί.

2.4 Μοντελοποίηση του προβλήματος V.R.P

Το απλό πρόβλημα V.R.P. είναι ένα πρόβλημα ακέραιου γραμμικού προγραμματισμού. Δίνεται ένα γράφημα G(V,E). Οι κόμβοι V=(1,2,...,N) αντιπροσωπεύουν τους πελάτες και την αποθήκη (όπου η αποθήκη είναι ο κόμβος 1). Ο αριθμός των οχημάτων έστω ότι είναι m. Επιπλέον δίδεται nxn πίνακας (c_{ij}) όπου c_{ij} είναι το κόστος μεταφοράς από τον κόμβο i στον κόμβο j. Η ζήτηση του κόμβου i είναι d_i (η ζήτηση του κόμβου 1 δηλαδή της αποθήκης είναι d_1=0). Κάθε όχημα έχει χωρητικότητα Q. Ζητείται να βρεθεί ένας πίνακας NxNxM $(X=x_{ijk})$, ώστε να ελαχιστοποιείται η συνάρτηση:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} c_{ij} x_{ijk}$$

υπό τους περιορισμούς:

$$\sum_{j=1}^{n} \sum_{k=1}^{m} x_{1jk} = m \qquad (1)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ijk} d_i \leq Q \qquad \text{για κάθε } k \in 1, \dots, m \qquad (2)$$

$$\sum_{i=1}^{n}\sum_{k=1}^{m} x_{ijk} = 1 \qquad \gamma\iota\alpha\ \kappa\acute{\alpha}\theta\varepsilon\ j \in 2, \dots, n \qquad (3)$$

$$\sum_{j=1}^{n} x_{ijk} - \sum_{j=1}^{n} x_{jik} = 0 \quad \gamma\iota\alpha\ \kappa\acute{\alpha}\theta\varepsilon\ i \in 1, \dots, n \quad \kappa\alpha\iota$$
$$\gamma\iota\alpha\ \kappa\acute{\alpha}\theta\varepsilon\ k \in 1, \dots, m \qquad (4)$$

$$\sum_{i,j \in S} x_{ijk} \leq |S| - 1 \qquad \gamma\iota\alpha\ \kappa\acute{\alpha}\theta\varepsilon\ k \in 1, \dots, m \ \ \kappa\alpha\iota$$
$$\gamma\iota\alpha\ \kappa\acute{\alpha}\theta\varepsilon\ S \subseteq \{2, \dots, n\} \qquad (5)$$

$$x_{ijk} \in \{0,1\} \qquad \gamma\iota\alpha\ \kappa\acute{\alpha}\theta\varepsilon\ k \in 1, \dots, m \ \ \kappa\alpha\iota$$
$$\gamma\iota\alpha\ \kappa\acute{\alpha}\theta\varepsilon\ i \in 1, \dots, n \quad \kappa\alpha\iota$$
$$\gamma\iota\alpha\ \kappa\acute{\alpha}\theta\varepsilon\ j \in 1, \dots, n \qquad (6)$$

όπου:

$$x_{ijk} = \begin{cases} 1, & A\nu\ \tau o\ \acute{o}\chi\eta\mu\alpha\ k\ \pi\acute{\alpha}\varepsilon\iota\ \sigma\tau o\nu\ \kappa\acute{o}\mu\beta o\ j\ \alpha\mu\acute{\varepsilon}\sigma\omega\varsigma\ \mu\varepsilon\tau\acute{\alpha}\ \tau o\nu\ \kappa\acute{o}\mu\beta o\ i \\ 0, & \alpha\lambda\lambda\iota\acute{\omega}\varsigma \end{cases}$$

Ο 1ος περιορισμός εκφράζει ότι από την αποθήκη φεύγουν m οχήματα. Ο περιορισμός (2) δείχνει ότι τα οχήματα έχουν χωρητικότητα ίση με Q. Οι περιορισμοί (3) και (4), εξασφαλίζουν ότι ο κάθε κόμβος επισκέπτεται μία μόνο φορά και ότι το όχημα που εισέρχεται σε έναν κόμβο είναι το ίδιο με εκείνο που εξέρχεται από αυτόν. Ο περιορισμός (5) αφαιρεί κάθε φορά μία διαδρομή που ολοκληρώνεται. Τέλος ο 6ος περιορισμός μας δείχνει ότι όλες οι μεταβλητές απόφασης είναι δυαδικές (0 ή 1). [3]

ΚΕΦΑΛΑΙΟ 3

Μεμετικός Αλγόριθμος Πολλαπλών Πληθυσμών – Νησιών (Island Memetic Algorithm)

3.1 Εισαγωγή

Ο αλγόριθμος που παρουσιάζεται σε αυτήν την εργασία είναι μία καινοτομία στο χώρο της συνδυαστικής βελτιστοποίησης. Ανήκει στις μεθευρετικές μεθόδους επίλυσης προβλημάτων συνδυαστικής βελτιστοποίησης. Βέβαια, εσωτερικά συνδυάζει αρμονικά ευρετικές και μεθευρετικές μεθόδους. Με τις μεθευρετικές προσεγγίζονται τα διάφορα τοπικά βέλτιστα και με τις ευρετικές μεθόδους εντοπίζονται με ακρίβεια.

Οι μεθευρετικές μέθοδοι χρησιμοποιούνται για την επίλυση των πιο πολύπλοκων προβλημάτων βελτιστοποίησης. Οι μεθευρετικές μέθοδοι έχουν ως κύριο στόχο την εξονυχιστική εξερεύνηση των διαστημάτων των πιθανών λύσεων. Η ποιότητα των λύσεων που επιτυγχάνονται με τις μεθόδους αυτές είναι πολύ καλύτερη από την ποιότητα που δίνουν οι απλές ευρετικές μέθοδοι. Στις μεθευρετικές μεθόδους κατατάσσονται οι γενετικοί και οι μεμετικοί αλγόριθμοι.

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 [1]:

1. Χρησιμοποιούν ένα αριθμό από επαναληπτικές δοκιμές
2. Περιλαμβάνουν ένα ή περισσότερους πράκτορες (π.χ. χρωμοσώματα)
3. Λειτουργούν (στην περίπτωση των πολύ – πρακτόρων) βάση ενός μηχανισμού συνεργασίας και ανταγωνισμού.
4. Περιλαμβάνουν διαδικασίες αυτό – τροποποιήσεων των ευρετικών παραμέτρων ή ακόμα και της αναπαράστασης του προβλήματος.

Τα χαρακτηριστικά των μεθευρετικών είναι τα εξής [1]:

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

3.2 Εξελίξεις στο χώρο του προβλήματος VRP, σχετικά με τους μεθευρετικούς αλγορίθμους

Δύο δεκαετίες πριν άρχισαν να εφαρμόζονται μεθευρετικές προσεγγίσεις σε προβλήματα VRP. Μία από τις πρώτες προσεγγίσεις τέτοιου είδους είναι ο γενετικός αλγόριθμος που προτάθηκε από Ochi et al. [4] για το πρόβλημα VRP (μεταβλητού στόλου με σταθερά κόστη) όπου δημιουργείται ένας αρχικός πληθυσμός λύσεων με την ευρετική μέθοδο SWAP (η οποία περιγράφεται αργότερα). Ο ίδιος αλγόριθμος δοκιμάστηκε με παράλληλο προγραμματισμό στη δημοσίευση Ochi et al. [5]. Και στις δύο δημοσιεύσεις όμως δεν αναφέρονται λεπτομέρειες για τις υπολογιστικές δοκιμές.

Για αυτή την οικογένεια προβλημάτων αναπτύχθηκαν προσεγγίσεις της μεθόδου “Tabu Search” από τους Osman και Salhi [7], Gendreau et al. [8], και Wassan και Osman [9]. Όλοι αυτοί οι αλγόριθμοι ήταν επεκτάσεις προηγούμενων προσεγγίσεων για προβλήματα VRP (μεταβλητού στόλου με σταθερά κόστη αλλά και ετερογενή προβλήματα VRP με μεταβαλλόμενα κόστη ανάλογα με τα οχήματα), χρησιμοποιώντας έλεγχο εφικτότητας για κάθε ένα πρόβλημα και εκτίμηση της αντικειμενικής συνάρτησης. Συγκεκριμένα οι Osman και Salhi [7] χρησιμοποίησαν μία και μόνο ανταλλαγή μεταξύ γειτονικών λύσεων μαζί με ένα απλό μηχανισμό λίστας “tabu”, ενώ οι Wassan και Osman [9] ανέμειξαν διάφορες αποδοτικές στρατηγικές για να βελτιώσουν την ποιότητα του αλγορίθμου. Διάφοροι ευαίσθητοι μηχανισμοί αναζήτησης μεταξύ “γειτονικών” λύσεων βασισμένοι σε λ ανταλλαγές μεταξύ τους συνδυάζονται με αποδοτικές τεχνικές διαχείρισης δεδομένων για το χειρισμό των λιστών “tabu”. Η μέθοδος “tabu search” στη δημοσίευση Gendreau et al. [8], περιλαμβάνει έναν κλασικό αλγόριθμο βασισμένο σε “γείτονες με καλά χαρακτηριστικά” με προσαρμόσιμους μηχανισμούς μνήμης των Rochat and Taillard [10]. Τα αποτελέσματα του αλγορίθμου των Osman και Salhi [7] (στα παραδείγματα G20) είχαν ένα μέσο ποσοστιαίο σφάλμα περίπου 0,68% με βάση τις καλύτερες μέχρι τότε δημοσιευμένες λύσεις. Ο αλγόριθμος από Gendreau et al. [8] δοκιμάστηκε στα παραδείγματα G12, έχοντας ένα μέσο ποσοστιαίο σφάλμα ίσο με 0,24%, με μέσο χρόνο υπολογισμού 765 δευτερόλεπτα σε ένα Sun Sparcstation 10. Ο ίδιος αλγόριθμος δοκιμάστηκε επίσης στα παραδείγματα T8, έχοντας ένα μέσο ποσοστιαίο σφάλμα ίσο με 0.09% με μέσο χρόνο υπολογισμού ίσο με 1151 δευτερόλεπτα. Η μέθοδος ‘tabu search’ των Wassan και Osman [9] παράγει καλά αποτελέσματα στα παραδείγματα G20 καθώς και στα παραδείγματα T8. Συγκεκριμένα στα παραδείγματα G20, το μέσο ποσοστιαίο σφάλμα είναι ίσο με 0.41% και ο μέσος χρόνος υπολογισμού είναι ίσος με 1215 δευτερόλεπτα. Στα παραδείγματα T8, το μέσο ποσοστιαίο σφάλμα είναι ίσο με 0.47% και ο μέσος χρόνος υπολογισμού είναι ίσος με 2098 δευτερόλεπτα σε έναν Sun Sparc 1000.

Επίσης έχουν αναπτυχθεί δύο μεθευρετικοί αλγόριθμοι για το πρόβλημα VRP (ετερογενές VRP με κόστη που εξαρτώνται από τα οχήματα) από Tarantilis et al. [11, 12] βασισμένοι στην αποδοχή των λύσεων με βάση ένα κατώφλι: και οι δύο μέθοδοι ξεκινούν από μία λύση προερχόμενη από έναν ευρετικό αλγόριθμο, και στην συνέχεια ακολουθείται μία επαναληπτική διαδικασία αποδοχής με βάση τα κατώφλια. Σε αυτή τη διαδικασία, δημιουργείται επαναληπτικά μία τυχαία λύση στη “γειτονιά” της τρέχουσας λύσης επίσης υπολογίζεται και η τιμή για το κατώφλι η οποία αναπαριστά τη βελτίωση σε σχέση με την τιμή της τρέχουσας λύσης. Η νέα λύση μπορεί να γίνει αποδεκτή μετά από μία διαδικασία σύγκρισης της τιμής του κατωφλίου της με τις τιμές των κατωφλίων των M μέχρις στιγμής καλύτερων λύσεων (οι οποίες είναι αποθηκευμένες σε μία λίστα). Σε αυτές τις δύο δημοσιεύσεις έχουν προταθεί διαφορετικοί τρόποι για την ανανέωση της λίστας με τα κατώφλια. Οι μέθοδος στην Tarantilis et al. [11] δοκιμάστηκε στα παραδείγματα T8 και παρήγαγε αποτελέσματα με μέση απόκλιση 0,79% σε σχέση με τις μέχρις τότε καλύτερες λύσεις με μέσο χρόνο υπολογισμού τα 223 δευτερόλεπτα σε υπολογιστή Pentium III, των 550 MHz. Τα αποτελέσματα της μεθόδου από Tarantilis et al. [12] είναι ελαφρώς καλύτερα στα παραδείγματα T8 με μέση απόκλιση 0,62% και μέσο χρόνο υπολογισμού 607 δευτερόλεπτα σε υπολογιστή Pentium II/400. Στην δημοσίευση Li et al. [13] προτείνεται μια παρόμοια προσέγγιση η οποία είναι παραλλαγή της μεθευρετικής προσομοιωμένης ανόπτησης. Η μέθοδός τους δοκιμάστηκε στην δημοσίευση Golden et al. [14] στα παραδείγματα T8 και σε πενταβάθμια παραδείγματα 200 έως 360 πελατών. Στα προβλήματα T8, πέτυχαν μέση ποσοστιαία απόκλιση ίση με 0,03% με μέσο χρόνο υπολογισμού 286 δευτερόλεπτα σε έναν υπολογιστή Athlon 1 GHz.

.

Επίσης προτείνεται μια προσέγγιση "αποσύνθεσης-σύνθεσης" για το πρόβλημα VRP μεταβλητού στόλου με σταθερά κόστη και χρονικά παράθυρα στη δημοσίευση Dell'Amico et al. [15]. Συγκεκριμένα, χρησιμοποιείται μια παράλληλη διαδικασία εισαγωγής για την παροχή μιας αρχικής λύσης και για την πιθανή συμπλήρωση των μερικών λύσεων που παράγονται στο βήμα της "αποσύνθεσης". Αυτό το βήμα πραγματοποιείται επιλέγοντας μία διαδρομή για να καταργηθεί. Η προτεινόμενη προσέγγιση υπερτερούσε από τους αλγορίθμους των Liu και Shen [16] και Dullaert et al. [17] όσον αφορά στα παραδείγματα LS168. Στη δημοσίευση Braysy et al. [17] προτείνεται μία νέα ντετερμινιστική μεθευρετική ανόπτηση για το πρόβλημα VRP μεταβλητού στόλου με σταθερά κόστη και χρονικά παράθυρα. Τέτοιοι μεθευρετικοί αλγόριθμοι χωρίζονται σε τρεις φάσεις. Καταρχήν δημιουργούνται αρχικές λύσεις με ευρετικούς αλγορίθμους που συνδυάζουν τεχνικές διαφοροποίησης και "μηχανισμούς εκμάθησης", Στη συνέχεια γίνεται μία προσπάθεια για τη μείωση των διαδρομών της αρχικής λύσης με μία εκ νέου διαδικασία τοπικής αναζήτησης και τέλος η λύση της δεύτερης φάσης βελτιώνεται ακόμα περισσότερο από ένα σετ τεσσάρων μεθόδων τοπικής αναζήτησης που ενσωματώνονται στα πλαίσια μίας ντετερμινιστικής ανόπτησης. Τα υπολογιστικά πειράματα στα δοκιμαστικά παραδείγματα LS168 δείχνουν ότι αυτή η προτεινόμενη μέθοδος υπερτερεί των αποτελεσμάτων που είχαν δημοσιευθεί νωρίτερα και μάλιστα βελτιώνει σχεδόν όλες τις μέχρι τότε γνωστές βέλτιστες λύσεις.

Υπάρχουν δύο μεθευρετικές μέθοδοι για το πρόβλημα VRP όπου οι τύποι των οχημάτων εξαρτώνται από τους πελάτες που εξυπηρετούν. Αυτές οι δύο μεθευρετικές μέθοδοι βασίζονται στην αναγωγή του προβλήματος σε γενικό πρόβλημα διαδρομών. Η πρώτη μέθοδος είναι η "tabu search" που προτάθηκε από τους Cordeau and Laporte [20] για την παραλλαγή του προβλήματος VRP με "χρονικά παράθυρα". Το πρόβλημα ανάγεται σε ένα πρόβλημα με χρονικές περιόδους όπου κάθε τύπος οχήματος σχετίζεται με μία διαφορετική περίοδο-μέρα και επιβάλλεται ότι κάθε πελάτης πρέπει να εξυπηρετηθεί μία φορά σε μία από τις μέρες που αντιστοιχούν στο συμβατό όχημα. Στη συνέχεια το πρόβλημα λύνεται χρησιμοποιώντας τη μέθοδο "tabu search" που προτείνεται στην δημοσίευση Cordeau et al. [21]. Η μέση ποσοστιαία απόκλιση ισούται με 1,48% με μέσο χρόνο υπολογισμού τα 12 δευτερόλεπτα σε έναν Sun Ultra 2 (300 MHz). Ο δεύτερος μεθευρετικός αλγόριθμος για το πρόβλημα VRP όπου οι τύποι των οχημάτων εξαρτώνται από τους πελάτες που εξυπηρετούν, δημιουργήθηκε από τους Pisinger και Ropke [22]. Μετέτρεψαν το πρόβλημα σε ένα πρόβλημα "συλλογής-διανομής" με χρονικά παράθυρα, το οποίο λύθηκε χρησιμοποιώντας ένα προσαρμοστικό αλγόριθμο αναζήτησης . Η μέθοδος τους υπερέχει ξεκάθαρα όλως των προηγούμενων που περιγράφηκαν πριν, όσον αφορά στα αποτελέσματα και ο μέσος χρόνος υπολογισμού ήταν 162 δευτερόλεπτα σε έναν υπολογιστή Pentium 4, των 3 GHz. [18]

Επίσης οι βέλτιστες λύσεις που έχουν βρεθεί έως τώρα (σε παγκόσμια κλίμακα) για τα παραδείγματα που χρησιμοποιήσαμε σε αυτήν την εργασία έχουν ως εξής: [19]

ΠΑΡΑΔΕΙΓΜΑ	ΜΕΧΡΙ ΣΤΙΓΜΗΣ ΒΕΛΤΙΣΤΟ
1^{o} παράδειγμα	524,61
2^{o} παράδειγμα	835,26
3^{o} παράδειγμα	826,14
4^{o} παράδειγμα	1028,42
5^{o} παράδειγμα	1291,29
6^{o} παράδειγμα	555,43
7^{o} παράδειγμα	909,68
8^{o} παράδειγμα	865,94
9^{o} παράδειγμα	1162,55
10^{o} παράδειγμα	1395,85
11^{o} παράδειγμα	819,56
12^{o} παράδειγμα	1042,11
13^{o} παράδειγμα	866,37
14^{o} παράδειγμα	1541,14

3.3 Ο Μεμετικός Αλγόριθμος στο πρόβλημα VRP

3. 3.1 Εισαγωγή

Ο Μεμετικός Αλγόριθμος ανήκει στις μεθευρετικές μεθόδους βελτιστοποίησης. Είναι μια παραλλαγή του Γενετικού Αλγορίθμου και διαφέρει από αυτόν μονάχα στο ότι στον Μεμετικό αλγόριθμο εκτός από τις διαδικασίες του Γενετικού Αλγορίθμου χρησιμοποιούμε και τοπική αναζήτηση.

3.3.2 Ο Γενετικός Αλγόριθμος

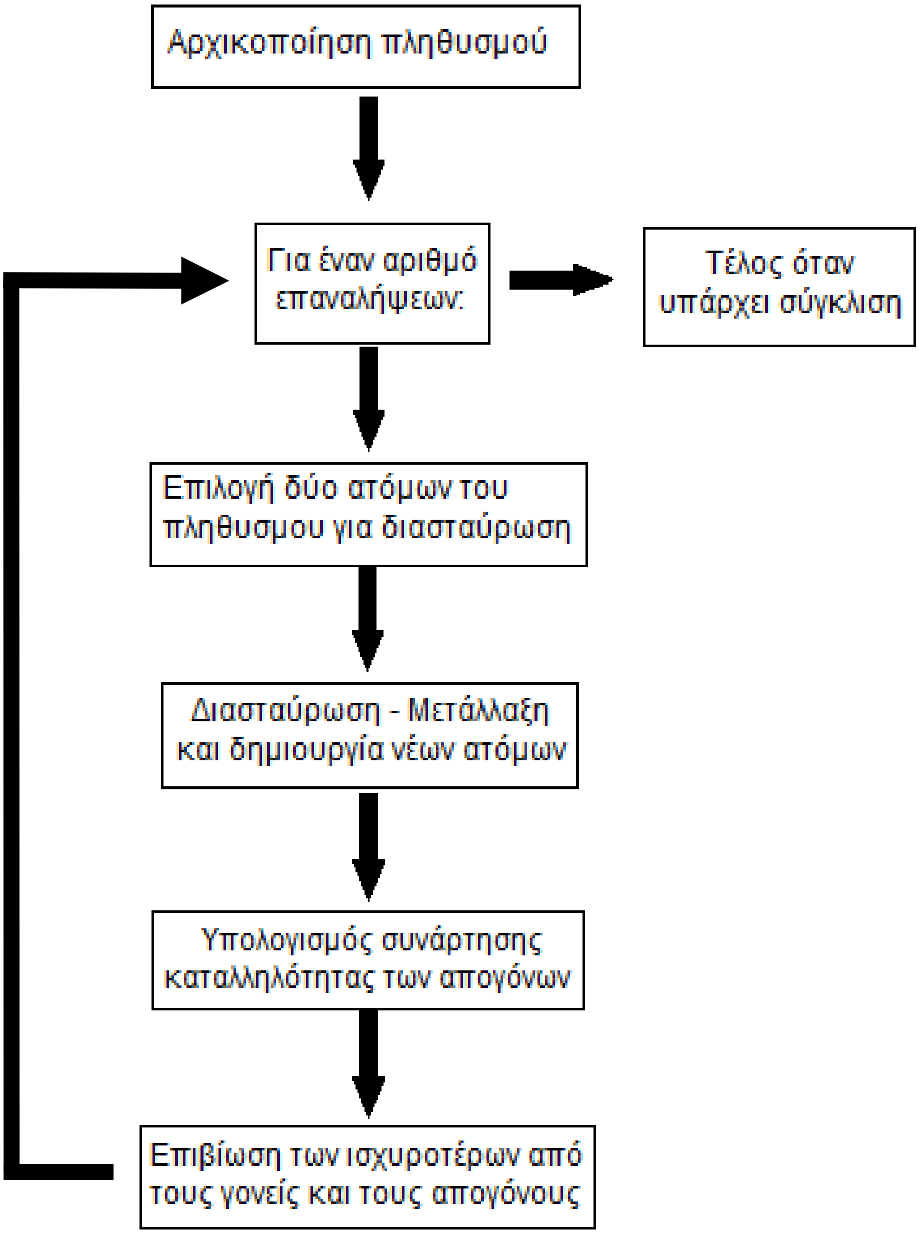
Οι γενετικοί αλγόριθμοι μιμούνται την φυσική διαδικασία της εξέλιξης των ειδών όπου αναπαράγονται. Κατά τη διάρκεια της αναπαραγωγής, η γονιδιακή δομή του απογόνου καθορίζεται, κατά μέσο όρο, κατά το ήμισυ από τη γονιδιακή δομή του ενός γονέα και το άλλο μισό καθορίζεται από την γονιδιακή δομή του άλλου γονέα. Τελικά, η γονιδιακή δομή του απογόνου μεταλλάσσεται. Εάν ο απόγονος κληρονομήσει καλά χαρακτηριστικά, η πιθανότητα επιβίωσης του είναι υψηλότερη και υπάρχει μεγαλύτερη πιθανότητα να γίνει γονέας και να συντελέσει στην παραγωγή νέων απογόνων. Έτσι, τα καλά χαρακτηριστικά μεταβιβάζονται ευκολότερα κι ο πληθυσμός τείνει να βελτιώνεται με την πάροδο του χρόνου. Έτσι οι γενετικοί αλγόριθμοι αποτελούν μία πολύ αποτελεσματική μέθοδο για την επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης μεγάλης κλίμακας όπως το VRP.

Οι λύσεις διαδραματίζουν το ρόλο των ατόμων, η διαδικασία αναπαραγωγής προσομοιάζεται από μία διασταύρωση που συνδυάζει δύο λύσεις για να δημιουργήσει μια νέα που αλλάζει τελικά μέσω της διαδικασίας της μετάλλαξης.

Με αυτόν τον τρόπο λοιπόν, οι γενετικοί αλγόριθμοι μιμούνται τη διαδικασία της εξέλιξης σε ένα πρόβλημα βελτιστοποίησης. Κάθε εφικτή λύση ενός προβλήματος αντιμετωπίζεται ως ένα άτομο του οποίου η φυσική ικανότητα αντιστοιχεί στην τιμή της αντικειμενικής συνάρτησης. Οι γενετικοί αλγόριθμοι συντηρούν έναν πληθυσμό χρωμοσωμάτων στον οποίο μεγαλύτερη πιθανότητα επιβίωσης έχουν τα καταλληλότερα χρωμοσώματα. Υπάρχει συστηματοποιημένη, αλλά παράλληλα και τυχαία, ανταλλαγή πληροφοριών μεταξύ δύο ατόμων για την παραγωγή καλύτερων απογόνων.

Ενισχύεται επίσης η ποικιλομορφία του πληθυσμού μέσω της τυχαίας ανταλλαγής μερικών γονιδίων μεταξύ των ατόμων (μετάλλαξη) ή με την εισαγωγή νέων ατόμων (μετανάστευση). Οι γενετικοί αλγόριθμοι εφαρμόζουν επανειλημμένα αυτές τις διαδικασίες μέχρι ο πληθυσμός να συγκλίνει. Οι γενετικοί αλγόριθμοι μπορούν να εφαρμοστούν με ποικίλους τρόπους. [1]

Σχηματικά η απλούστερη περιγραφή του γενετικού αλγορίθμου έχει ως εξής:



3.3.3 Ο Μεμετικός Αλγόριθμος

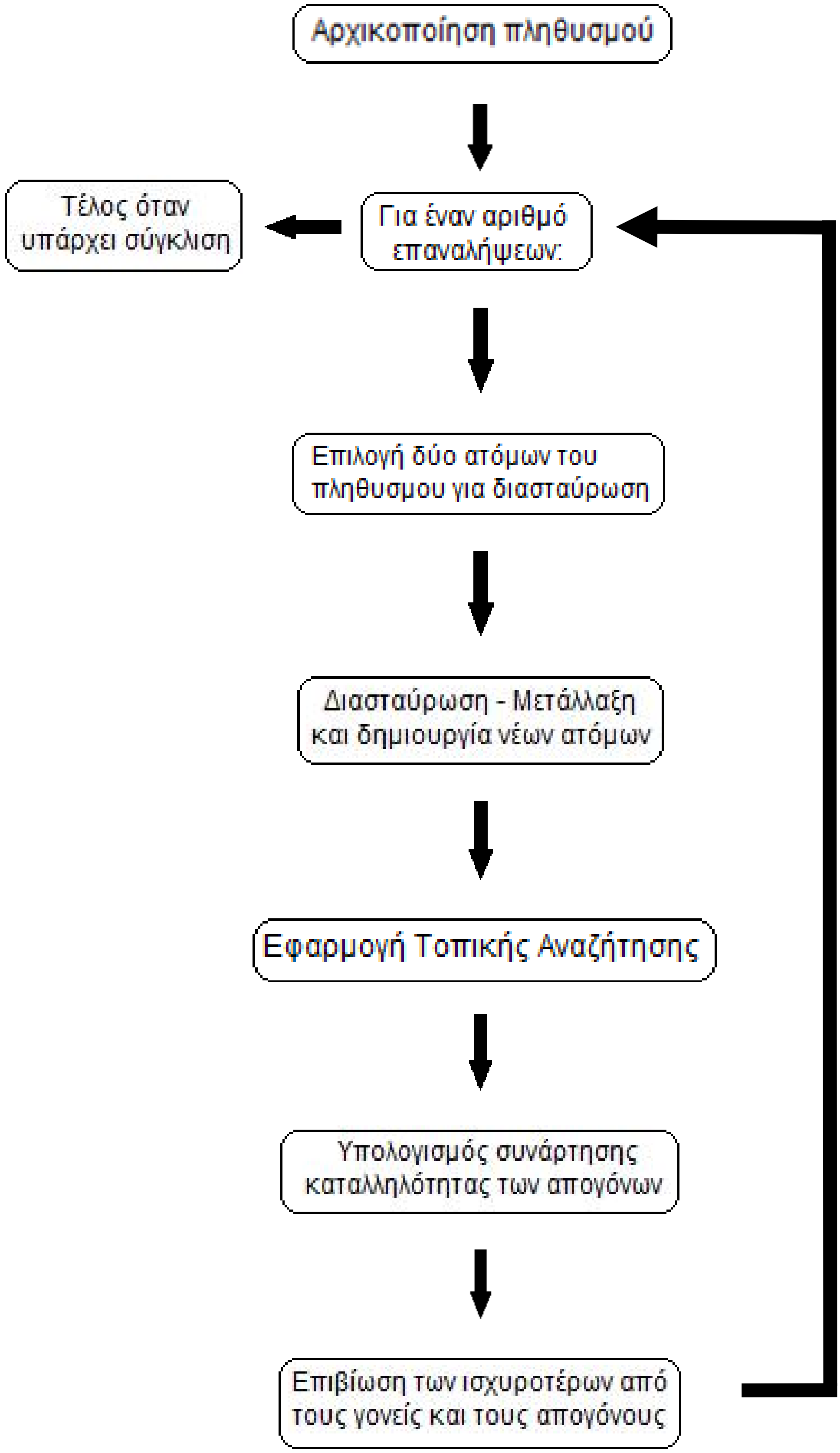
Ο Μεμετικός Αλγόριθμος έρχεται να καλύψει τα βασικά μειονεκτήματα του Γενετικού Αλγορίθμου. Ο Γενετικός Αλγόριθμος μετά από ένα αριθμό γενεών – βημάτων συγκλίνει σε κάποιο αποτέλεσμα, χωρίς αυτό το αποτέλεσμα να είναι απαραίτητα η βέλτιστη λύση ή έστω κάποια καλή λύση.

Σε αυτό το μειονέκτημα του Γενετικού Αλγορίθμου απαντάει ο Μεμετικός Αλγόριθμος χρησιμοποιώντας Τοπική Αναζήτηση εκτός από τις διαδικασίες του Γενετικού Αλγορίθμου. Στην ουσία ο Μεμετικός Αλγόριθμος επιβραδύνει τη σύγκλιση του Αλγορίθμου ωθώντας (περισσότερο από το Γενετικό) τον αλγόριθμο προς ένα τοπικό βέλτιστο. Έτσι ο Μεμετικός Αλγόριθμος δίνει τη δυνατότητα περεταίρω βελτίωσης μιας λύσης που θα έδινε ένας γενετικός αλγόριθμος. Εκτός όμως από το ότι ο Μεμετικός Αλγόριθμος βελτιώνει τις λύσεις του με τοπική αναζήτηση ωθώντας τον αλγόριθμο σε καλύτερα αποτελέσματα, αυτή η ίδια η βελτίωση των αποτελεσμάτων επιβραδύνει τη σύγκλιση με αποτέλεσμα ο αλγόριθμος να δίνει ακόμα καλύτερα αποτελέσματα.

Έτσι σε έναν Μεμετικό Αλγόριθμο χρησιμοποιούνται όλες οι διαδικασίες του γενετικού αλγορίθμου (Διασταύρωση, Μετάλλαξη κτλ) συν την τοπική αναζήτηση η οποία δεν χρησιμοποιείται στους γενετικούς αλγορίθμους.

Ο Μεμετικός Αλγόριθμος σχηματικά παρουσιάζεται στην επόμενη σελίδα :

Μεμετικός Αλγόριθμος:



3.3.4 Γενετικός Αλγόριθμος Πολλαπλών Πληθυσμών – Νησιών (Island Memetic Algorithms)

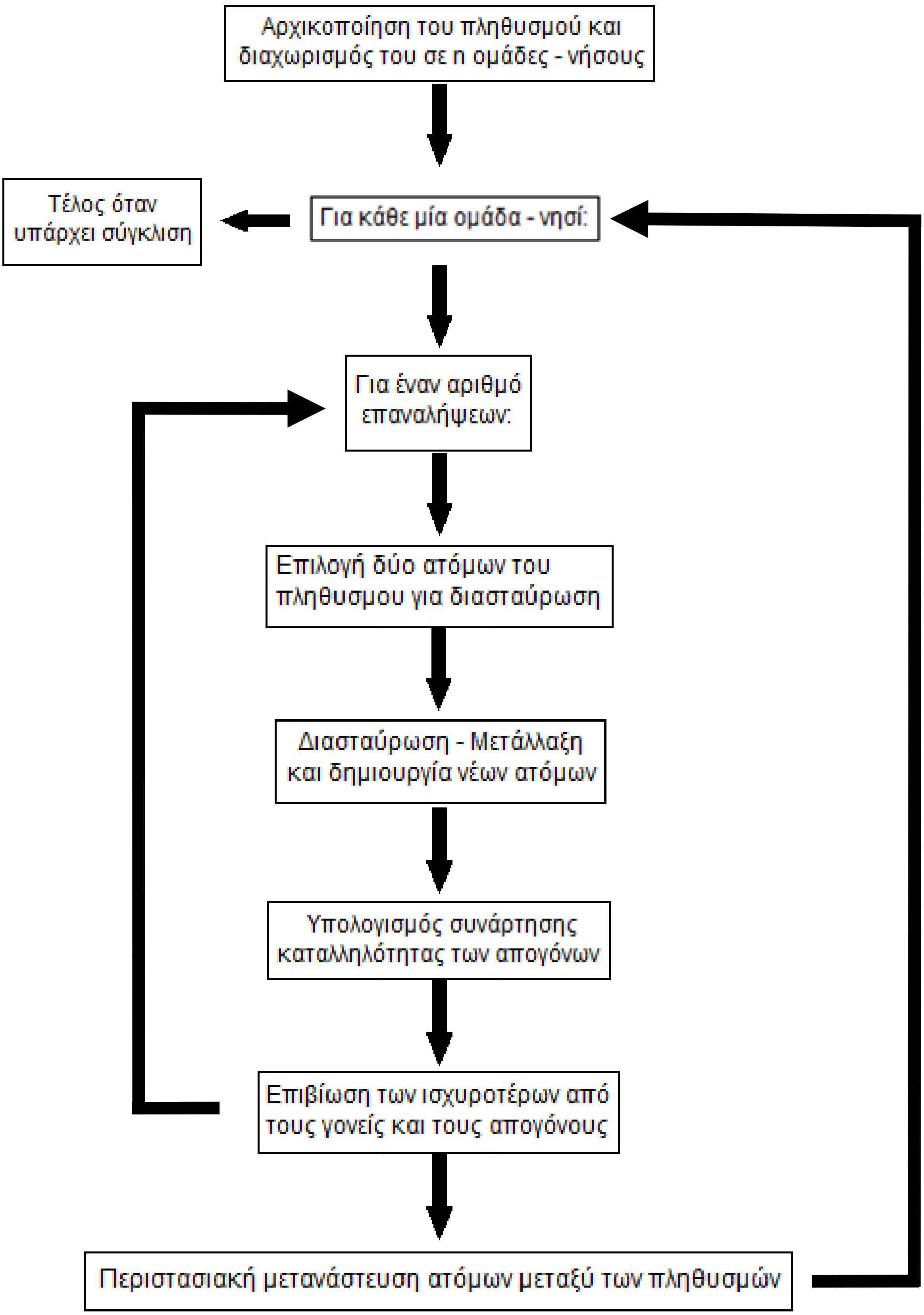
Οι Γενετικοί Αλγόριθμοι Πολλαπλών Πληθυσμών – Νησιών (Island Genetic Algorithms).είναι μια παραλλαγή των κλασικών αλγορίθμων, που συνήθως εφαρμόζονται σε παράλληλο προγραμματισμό. Οι πληθυσμοί εξελίσσονται συνεργαζόμενοι ο ένας με τον άλλο παράλληλα. Αρχικά ένας πληθυσμός από τυχαία άτομα δημιουργείται όπως ακριβώς και στους κλασικούς γενετικούς αλγόριθμους. Στην συνέχεια ο αρχικός πληθυσμός διαιρείται σε πολλαπλούς πληθυσμούς τα νησιά, βάση της επιλογής του χρήστη (ο αριθμός των νησιών είναι μια από τις παραμέτρους του αλγόριθμου). Κάθε νησί αντιπροσωπεύει ένα διαφορετικό πληθυσμό, όπου οι βασικοί τελεστές των γενετικών αλγορίθμων (επιλογή των γονέων, διασταύρωση και μετάλλαξη)εφαρμόζονται ξεχωριστά από τα άλλα νησιά. Η βασική διαφορά από τους γενετικούς αλγόριθμους είναι η ύπαρξη μιας πολιτικής μετανάστευσης (migration policy) κάθε πληθυσμού. Η μετανάστευση χρησιμοποιείται με στόχο την ανταλλαγή πληροφοριών ανάμεσα στα διαφορετικά νησιά. Έχουν προταθεί αρκετοί τρόποι για να πραγματοποιείται η μετανάστευση. Αρχικά πρέπει να καθοριστεί το ποσό του πληθυσμού που θα μεταναστεύσει σε άλλο νησί και πόσες φορές θα επαναληφθεί κατά τη διάρκεια των επαναλήψεων. Μια στρατηγική είναι να ακολουθηθεί η τοπολογία δακτυλίου, όπου οι μετανάστες μεταναστεύουν σε διπλανό νησί και αντικαθιστούν ένα μέρος του πληθυσμού. Όταν θα φτάσει στο διπλανό νησί θα πρέπει να δούμε ποιο μέλος του πληθυσμού θα αντικαταστήσει.

Οι δυνατότητες που συνήθως εμφανίζονται είναι τέσσερις [1]:

1. Ένας καλός μετανάστης αντικαθιστά ένα κακό μέρος του πληθυσμού,
2. Ένας καλός μετανάστης αντικαθιστά ένα τυχαία επιλεγμένο μέλος του πληθυσμού
3. Ένας τυχαία επιλεγμένος μετανάστης αντικαθιστά ένα κακό μέλος του πληθυσμού
4. Και ένας τυχαία επιλεγμένος μετανάστης αντικαθιστά ένα τυχαία επιλεγμένο μέλος του πληθυσμού.

Η μετανάστευση πραγματοποιείται όσες φορές επιλέξει ο χρήστης. Έτσι, αν ο χρήστης επιλέξει μικρό αριθμό από μεταναστεύσεις τότε υπάρχει μεγάλη πιθανότητα να μην έχει προλάβει να πραγματοποιηθεί σύγκλιση σε κάθε νησί και άρα να μεταφερθούν αρκετά μέτριοι μετανάστες από το ένα νησί στο άλλο.

Σχηματικά ο Γενετικός Αλγόριθμος Πολλαπλών Πληθυσμών – Νησιών (Island Genetic Algorithm) έχει ως εξής:



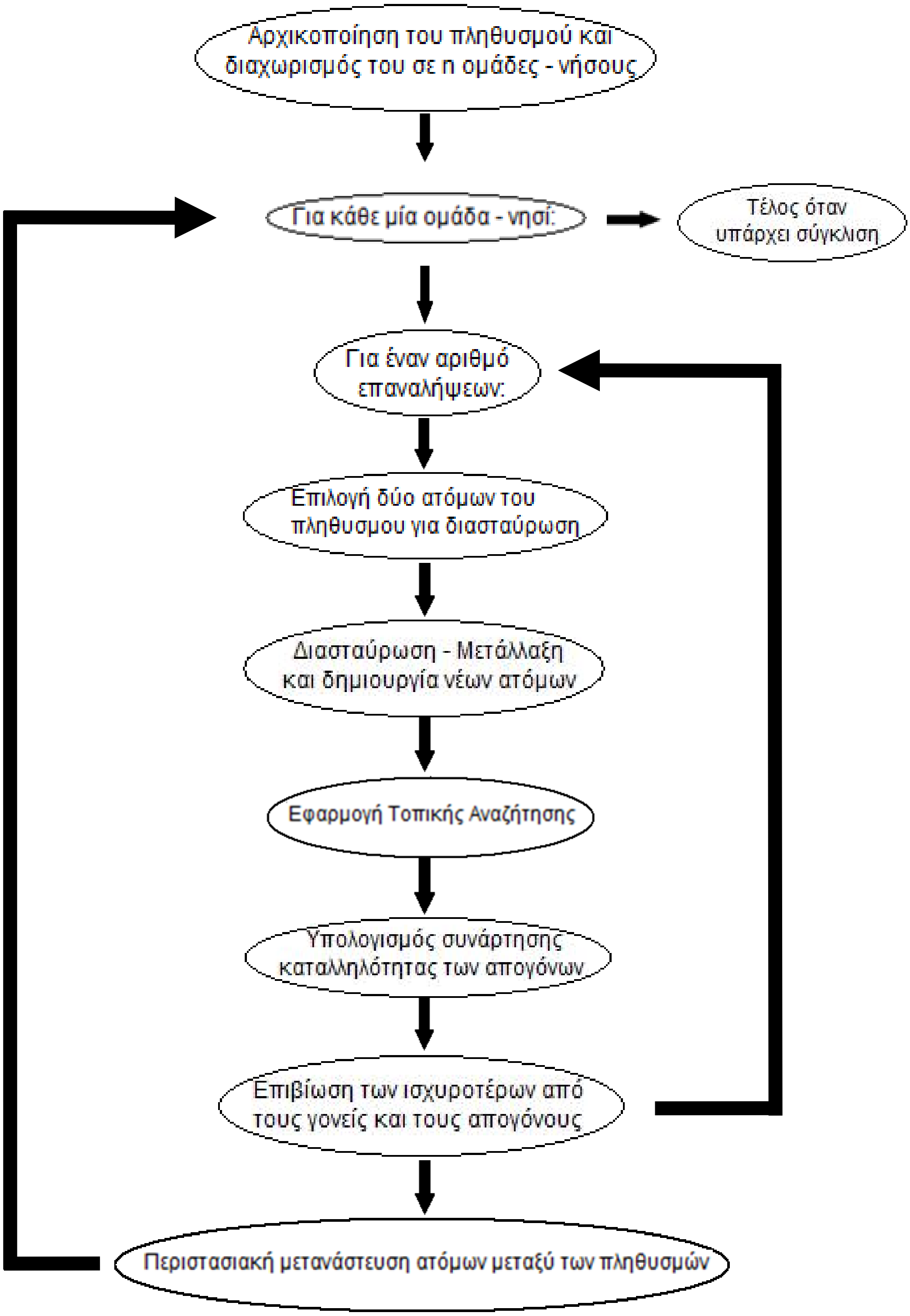
3.3.5 Μεμετικός Αλγόριθμος Πολλαπλών Πληθυσμών – Νησιών (Island Memetic Algorithm)

Ο Μεμετικός Αλγόριθμος Πολλαπλών Πληθυσμών – Νησιών (Island . Memetic Algorithm) προτείνεται για πρώτη φορά από αυτήν την εργασία έρχεται να καλύψει τα βασικά μειονεκτήματα του Γενετικού Αλγορίθμου Πολλαπλών Πληθυσμών – Νησιών (Island Genetic Algorithm). Ανήκει στις μεθευρετικές μεθόδους επίλυσης προβλημάτων συνδυαστικής βελτιστοποίησης. Συνδυάζει αρμονικά ευρετικές και μεθευρετικές μεθόδους επίλυσης. Με τη βοήθεια των μεθευρετικών μεθόδων προσεγγίζονται τα διάφορα τοπικά βέλτιστα και με τις ευρετικές μεθόδους εντοπίζονται με ακρίβεια. Ο Γενετικός Αλγόριθμος Πολλαπλών Πληθυσμών – Νησιών μετά από ένα αριθμό γενεών (βημάτων) συγκλίνει σε κάποιο αποτέλεσμα, χωρίς αυτό το αποτέλεσμα να είναι απαραίτητα η βέλτιστη λύση ή έστω κάποια καλή λύση. Σε αυτό το μειονέκτημα του Γενετικού Αλγορίθμου Πολλαπλών Πληθυσμών – Νησιών απαντάει ο Μεμετικός Αλγόριθμος χρησιμοποιώντας Τοπική Αναζήτηση εκτός από τις διαδικασίες του Γενετικού Αλγορίθμου Πολλαπλών Πληθυσμών – Νησιών.

Στην ουσία ο Μεμετικός Αλγόριθμος Πολλαπλών Πληθυσμών – Νησιών επιβραδύνει τη σύγκλιση του Αλγορίθμου ωθώντας (περισσότερο από το Γενετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών) τον αλγόριθμο προς ένα τοπικό βέλτιστο. Έτσι ο Μεμετικός Αλγόριθμος Πολλαπλών Πληθυσμών – Νησιών δίνει τη δυνατότητα περεταίρω βελτίωσης μιας λύσης που θα έδινε ένας Γενετικός Αλγόριθμος Πολλαπλών Πληθυσμών – Νησιών. Εκτός όμως από το ότι ο Μεμετικός Αλγόριθμος Πολλαπλών Πληθυσμών – Νησιών βελτιώνει τις λύσεις του με τοπική αναζήτηση ωθώντας τον αλγόριθμο σε καλύτερα αποτελέσματα, αυτή η ίδια η βελτίωση των αποτελεσμάτων επιβραδύνει τη σύγκλιση με αποτέλεσμα ο αλγόριθμος να δίνει ακόμα καλύτερα αποτελέσματα.

Έτσι σε έναν Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών χρησιμοποιούνται όλες οι διαδικασίες του γενετικού αλγορίθμου Πολλαπλών Πληθυσμών – Νησιών (Διασταύρωση, Μετάλλαξη, Μετανάστευση κτλ) συν την τοπική αναζήτηση η οποία δεν χρησιμοποιείται στους Γενετικούς Αλγορίθμους Πολλαπλών Πληθυσμών – Νησιών. Η τοπική αναζήτηση μπορεί να εφαρμοστεί είτε μετά είτε πριν τη μετανάστευση είτε και στα δύο σημεία.

Σχηματικά ο Μεμετικός Αλγόριθμος Πολλαπλών Πληθυσμών – Νησιών (Island Memetic Algorithm) έχει ως εξής:



3.4 Υλοποίηση του Μεμετικού Αλγορίθμου Πολλαπλών Πληθυσμών – Νησιών (Island Memetic Algorithm)

Καταρχήν αξίζει να σημειωθεί ότι στον αλγόριθμό μας οι λύσεις αναπαριστώνται με διάνυσμα της μορφής [… i j y …], το οποίο μας δείχνει την σειρά που επισκεπτόμαστε τους κόμβους, όπου ο αριθμός i αντιπροσωπεύει τον i-οστό κόμβο, ο αριθμός j αντιπροσωπεύει τον j-οστό κόμβο και ο αριθμός y αντιπροσωπεύει τον y-οστό κόμβο.

Στην αρχή φορτώνονται όλα τα δεδομένα του προβλήματος από ένα αρχείο excel. Μέσα στα δεδομένα που φορτώνονται υπάρχουν και οι συντεταγμένες του κάθε κόμβου. Βάσει αυτών των συντεταγμένων αποθηκεύουμε τις αποστάσεις από κόμβο σε κόμβο σε έναν πίνακα διάστασης $n \times n$, όπου n είναι ο αριθμός των κόμβων. Η απόσταση μεταξύ δύο κόμβων βρίσκεται υπολογίζοντας την ευκλείδεια απόσταση μεταξύ τους.

Στην συνέχεια δημιουργούμε κάποιες αρχικές λύσεις με την μέθοδο της Άπληστης Τυχοποιημένης Προσαρμοστικής Αναζήτησης (Greedy Randomized Adaptive Search Procedure – GRASP).

3.4.1 Άπληστη Τυχοποιημένη Προσαρμοστική Αναζήτηση (Greedy Randomized Adaptive Search Procedure – GRASP)

3.4.1.1 Περιγραφή της μεθόδου

Η Άπληστη Τυχοποιημένη Προσαρμοστική Αναζήτηση (Greedy . Randomized Adaptive Search Procedure – GRASP) είναι μια επαναληπτική διαδικασία και χρησιμοποιείται για την εύρεση προσεγγιστικών λύσεων. Η συγκεκριμένη μέθοδος παρέχει μια εφικτή λύση σε κάθε επανάληψη και συνεχίζεται έως ότου φτάσουμε σε κάποιο σημείο όπου η λύση δεν μπορεί να βελτιωθεί άλλο είτε όταν πραγματοποιηθεί ο μέγιστος αριθμός επαναλήψεων είτε όταν ικανοποιηθεί κάποιο άλλο κριτήριο λήξης. Το τελικό αποτέλεσμα είναι απλά η καλύτερη λύση σε όλες τις επαναλήψεις. Κάθε επανάληψη αποτελείται από δυο φάσεις, μια φάση κατασκευής μιας αρχικής λύσης (construction phrase) και μια διαδικασία τοπικής αναζήτησης (local search phrase) για βελτιστοποίηση αυτής της λύσης . Στην φάση κατασκευής, μια τυχοποιημένη συνάρτηση απληστίας χρησιμοποιείται για την κατασκευαστεί μια αρχική λύση. Αυτή η αρχική λύση στη συνέχεια βελτιώνεται με τη χρήση της διαδικασίας τοπικής αναζήτησης.

Η φάση κατασκευής μπορεί να περιγραφεί σαν η επαναληπτική πρόσθεση ενός στοιχείου στην μη ολοκληρωμένη λύση σε κάθε επανάληψη. Η στρατηγική επιλογής του επόμενου στοιχείου βασίζεται στην τυχαία επιλογή από μια λίστα υποψηφίων, που ονομάζεται λίστα περιορισμού των υποψηφίων (Restricted Candidate List) για εισαγωγή στην λύση στην οποία κάθε στοιχείο κατατάσσεται βάση μιας συνάρτησης απληστίας. Η τυχαία επιλογή του στοιχείου σημαίνει ότι δεν είναι ανάγκη να επιλέγει το πρώτο στοιχείο στη λίστα. Ο ευρετικός αλγόριθμος είναι προσαρμοστικός με την έννοια ότι η συνάρτηση επιλογής προσαρμόζεται για να προσμετρήσει τα στοιχεία που έχουν ήδη επιλεγεί .

Η λύση που δημιουργείται στην φάση κατασκευής δεν εγγυάται ότι είναι τοπικό ελάχιστο, και για αυτό το λόγο μια φάση τοπικής αναζήτησης εφαρμόζεται για να παράγει μια λύση που να περιέχει τοπικό ελάχιστο. Για να εφαρμοστεί η φάση τοπικής αναζήτησης, καθορίζεται μια συνάρτηση που κάνει αναζήτηση στη γειτονιά της αρχικής λύσης. Φυσικά, διαφορετικά προβλήματα χρειάζονται διαφορετικές συναρτήσεις απληστίας, διαφορετικές διαδικασίες τοπικής αναζήτησης για την εύρεση του τοπικού ελάχιστου όπως και διαφορετικές συναρτήσεις γειτονιάς που θα γίνει η αναζήτηση, αλλά όταν αυτά καθοριστούν τότε το μόνο που μας ενδιαφέρει για αν εξετάσουμε τον GRASP είναι το μέγεθος της λίστας και τον αριθμό των επαναλήψεων που χρειάζονται. [1]

3.4.1.2 Υλοποίηση της Άπληστης Τυχοποιημένης Προσαρμοστικής Αναζήτησης (GRASP)

Για την υλοποίηση της συγκεκριμένης μεθόδου χρησιμοποιούμε δεδομένα που αφορούν τη ζήτηση του κάθε κόμβου, τους χρόνους διαδρομών και εξυπηρέτησης κτλ. Τέτοιου είδους δεδομένα μας χρειάζονται για να υπολογίσουμε πότε ένα φορτηγό δεν μπορεί να επισκεφθεί κάποια άλλη πόλη λόγω των χωρικών ή χρονικών περιορισμών όπου υφίστανται. Χωρικούς περιορισμούς εννοούμε αυτούς που αναφέρονται στην χωρητικότητα των οχημάτων μεταφοράς π.χ. φορτηγών.

Έστω το διάνυσμα μίας λύσης που αναπαριστάται ως εξής: [… i j y …]. Υπάρχει η πιθανότητα το φορτηγό να μην μπορεί να πάει από τον κόμβο j στον κόμβο y, λόγω χρονικών ή χωρικών περιορισμών. Σε αυτήν την περίπτωση το φορτηγό επιστρέφει στον κόμβο 1 (στην αποθήκη δηλαδή), και στον κόμβο y θα πάει κάποιο άλλο φορτηγό αμέσως μετά την εκκίνησή του από τον κόμβο 1.

Έτσι ξεκινάμε να δημιουργούμε μια λύση ψάχνοντας στην αρχή τους πιο κοντινούς κόμβους από τον κόμβο έναρξης 1 (την αποθήκη δηλαδή). Στη συνέχεια επιλέγουμε έναν από αυτούς τους κόμβους για προορισμό του φορτηγού και τον προσθέτουμε στην πρώτη θέση του διανύσματος της λύσης. Έστω ότι επιλέξαμε τον κόμβο x και τον προσθέσαμε στην πρώτη θέση του διανύσματος της λύσης. Στο επόμενο βήμα από τους κόμβους που δεν έχουμε επισκεφθεί, εντοπίζουμε τους πιο κοντινούς ως προς τον κόμβο x. Εάν οι χωρικοί και χρονικοί περιορισμοί το επιτρέπουν τότε το φορτηγό θα επισκεφθεί κάποιον από αυτούς τους πιο κοντινούς κόμβους, τον οποίο θα τοποθετήσουμε στην δεύτερη θέση του διανύσματος της λύσης που κατασκευάζουμε, αλλιώς είτε το φορτηγό θα επισκεφθεί οποιοδήποτε άλλο κόμβο που επιτρέπουν οι περιορισμοί, είτε θα επιστρέψει στον κόμβο 1 (αποθήκη), από εκεί θα αναχωρήσει προς έναν σχετικά κοντινό κόμβο.

Αυτή η διαδικασία συνεχίζεται έως ότου σχηματιστεί ολόκληρο το διάνυσμα της λύσης, το οποίο στέλνουμε σαν όρισμα εξόδου προς το κύριο πρόγραμμα όπου θα χρησιμοποιηθεί για τη συνέχεια της διαδικασίας. Με αυτόν λοιπόν τον τρόπο ξεκινάμε από μία ομάδα καλών λύσεων. Αυτή η διαδικασία γίνεται μονάχα στην αρχή για κάθε μία ομάδα.

Για να υπολογίσουμε το κόστος της κάθε λύσης εργαζόμαστε όπως φαίνεται στην επόμενη σελίδα:

3.4.2 Υπολογισμός του κόστους

Για να υπολογίσουμε το συνολικό κόστος της λύσης αρχικά θα υπολογίσουμε το κόστος ταξιδίου από τον αρχικό κόμβο (αποθήκη) προς τον πρώτο κόμβο του διανύσματος της λύσης. Με δεδομένο ότι δεν έχουμε στα δεδομένα μας κάποια σταθερά κόστη και παίρνουμε υπόψη μας μόνο το κόστος ταξιδίου, το κόστος θα το υπολογίσουμε σε μονάδες απόστασης, όπου η απόσταση υπολογίζεται βάση των συντεταγμένων των κόμβων που μας δίνονται σαν δεδομένα.

Στη συνέχεια στο κόστος που υπολογίσαμε πριν θα προσθέσουμε το κόστος ταξιδίου από τον πρώτο κόμβο στον δεύτερο κόμβο του διανύσματος της λύσης, εάν βέβαια το επιτρέπουν οι χρονικοί περιορισμοί και οι περιορισμοί που αναφέρονται στη χωρητικότητα των οχημάτων μεταφοράς. Για να ελέγχουμε αυτούς τους περιορισμούς ανά πάσα στιγμή κάνουμε τα εξής: κάθε φορά που το φορτηγό επισκέπτεται μία πόλη προσθέτουμε σε μία μεταβλητή (έστω την d) τη ζήτηση του συγκεκριμένου κόμβου και σε μία άλλη μεταβλητή (έστω την x) το χρόνο εξυπηρέτησης του συγκεκριμένου κόμβου συν το χρόνο ταξιδίου του φορτηγού από τον προηγούμενο κόμβο σε αυτόν που πήγε τώρα (όπου και αυτός ο χρόνος μετράται σε μονάδες απόστασης λόγω της μορφής των δεδομένων που μας δίνονται).

Όταν οι περιορισμοί δεν ικανοποιούνται για τη μετακίνηση από έναν κόμβο (έστω τον i) στον επόμενό του (έστω τον j) τότε από τον κόμβο i το φορτηγό θα επιστρέψει στον κόμβο 1(αποθήκη) και στην μεταβλητή του κόστους προσθέτουμε το κόστος ταξιδίου του φορτηγού από τον κόμβο i στον κόμβο 1.

Σίγουρα ούτε τώρα (μετά την επιστροφή στον κόμβο 1) δεν θα υπερβαίνεται το χρονικό όριο που έχει τεθεί στον ένα περιορισμό αφού αυτή η περίπτωση είχε ελεγχθεί πριν επισκεφθούμε τον κόμβο i. Δηλαδή πριν το φορτηγό επισκεφθεί έναν κόμβο (έστω τον k) εκτός από τον έλεγχο που κάνουμε αν υπάρχει αρκετός χρόνος για να εξυπηρετήσει στον κόμβο k και να ταξιδέψει προς τον κόμβο k κάνουμε και τον έλεγχο αν υπάρχει αρκετός χρόνος για να επιστρέψει στον κόμβο 1 από τον κόμβο k, για την περίπτωση που χρειαστεί να γίνει αυτό μετά.

Μετά την επιστροφή λοιπόν στον κόμβο 1, μηδενίζουμε τις μεταβλητές x και d, οι οποίες μετρούν το χρόνο λειτουργίας του οχήματος μεταφοράς και τα προϊόντα που φορτώνονται σε αυτό. Μετά από τον κόμβο 1 μεταφερόμαστε στον κόμβο j προσθέτοντας στην μεταβλητή x το χρόνο εξυπηρέτησης του κόμβου j συν το χρόνο ταξιδίου του φορτηγού από τον κόμβο 1 στον κόμβο j. Επίσης προσθέτουμε στην μεταβλητή d τη ζήτηση του κόμβου j.

Αυτή η διαδικασία συνεχίζεται έως ότου φτάσουμε στον τελευταίο κόμβο του διανύσματος της λύσης (έστω τον z). Στην συνέχεια το όχημα μεταφοράς θα πρέπει να επιστρέψει από τον κόμβο z στον κόμβο 1(αποθήκη), άρα θα προσθέσουμε στο συνολικό κόστος το κόστος μεταφοράς από τον κόμβο j στον κόμβο 1.

Τώρα οι διαδικασίες που θα ακολουθήσουν επαναλαμβάνονται σε κάθε γενιά – βήμα. Αρχικά μπορούμε να βελτιώσουμε περεταίρω τις αρχικές λύσεις που δημιουργήσαμε (ή τις λύσεις που δίνονται από την προηγούμενη γενιά - βήμα) με Τοπική Αναζήτηση. Οι μέθοδοι Τοπικής Αναζήτησης που χρησιμοποιούμε σε αυτό το σημείο του αλγορίθμου είναι η μέθοδος 2-opt, η μέθοδος 3-opt, η μέθοδος Swap και η μέθοδος Relocate. Οι διαδικασίες Τοπικής Αναζήτησης ανήκουν στην ευρύτερη ομάδα των ευρετικών μεθόδων επίλυσης προβλημάτων συνδυαστικής βελτιστοποίησης.

3.4.3 Ευρετικές Μέθοδοι που χρησιμοποιούνται στον αλγόριθμό μας

3.4.3.1 Εισαγωγή

Όσο αυξάνει το μέγεθος του προβλήματος, η εύρεση της λύσης του γίνεται ολοένα και δυσκολότερη κυρίως λόγω του αυξημένου υπολογιστικού φόρτου. Γι αυτόν τον λόγο, σε μία ευρετική μέθοδο πραγματοποιείται μια σχετικά περιορισμένη αναζήτηση σε ένα διάστημα πιθανών λύσεων και ουσιαστικά παράγονται σχετικά γρήγορα λύσεις καλής ποιότητας.

Για να λύσουμε προβλήματα αυτής της μορφής συχνά καταφεύγουμε σε διάφορες τεχνικές που μας οδηγούν σε μια σχεδόν βέλτιστη, αλλά ικανοποιητική λύση. Μια λύση ενός ευρετικού αλγορίθμου γίνεται αποδεκτή αν ικανοποιεί κάποια κριτήρια όπως η ποιότητα της λύσης, δηλαδή η απόκλιση της από την βέλτιστη, η ευκολία απόκτησης μιας λύσης, η λογική πάνω στην οποία στηρίζονται οι κανόνες του ευρετικού αλγόριθμου που χρησιμοποιήθηκαν για να οδηγηθούμε στη λύση. Για κάποιο πρόβλημα συνδυαστικής βελτιστοποίησης δεν υπάρχει μονάχα ένας ευρετικός αλγόριθμος που να δίνει τη βέλτιστη λύση, αλλά έχουν αναπτυχθεί αρκετοί αλγόριθμοι οι οποίοι συγκρινόμενοι μεταξύ τους μας οδηγούν ολοένα και σε καλύτερες λύσεις. Ένα σημείο που πρέπει να διευκρινιστεί είναι αυτό που είπαμε προηγούμενα για την ποιότητα της λύσης. Σε μερικά προβλήματα, όπως είπαμε, είναι αδύνατο να βρούμε τη βέλτιστη λύση για κάποιο πρόβλημα σε ικανοποιητικό χρόνο, και έτσι γεννιέται το ερώτημα: Πως είναι δυνατό να είμαστε βέβαιοι για την ποιότητα της λύσης που πήραμε από τον αλγόριθμο που αναπτύξαμε; Η απάντηση δεν είναι εύκολο να δοθεί. Ένας απλός και ο πιο συνηθισμένος τρόπος απόδειξης είναι να δημιουργήσουμε μικρότερα παραδείγματα από αυτό που θέλουμε να λύσουμε, τα οποία μπορούμε να λύσουμε με κάποια ακριβή μέθοδο, και να δούμε πόσο κοντά στο βέλτιστο είναι η τιμή που παίρνουμε με τη χρήση του ευρετικού αλγορίθμου. Ένας άλλος τρόπος είναι η δημιουργία ενός φράγματος αποδεκτής λύσης με την επίλυση ενός χαλαρωμένου προβλήματος, για παράδειγμα με τη διαδικασία διακλάδωσης και οριοθέτησης (branch and bound),και έτσι όλες οι λύσεις που θα πάρουμε με τη χρήση του ευρετικού αλγορίθμου και που η τιμή τους δεν θα παραβιάζει την τιμή του φράγματος θα είναι ικανοποιητικές.

Υπάρχουν διάφορες κατηγορίες ευρετικών αλγορίθμων. Η κάθε μια από αυτές έχει κάποια ιδιαίτερα χαρακτηριστικά . Μερικές από τις κατηγορίες των αλγορίθμων αυτών είναι οι ακόλουθες είναι οι Αλγόριθμοι Απληστίας (Greedy Algorithms), Οι Προσεγγιστικοί Αλγόριθμοι (Approximation Algorithms) και Αλγόριθμοι Τοπικής Αναζήτησης (Local Search Algorithms).

Από τις ευρετικές μεθόδους, χρησιμοποιούμε αλγορίθμους τοπικής αναζήτησης όπως η μέθοδος 2-opt, η μέθοδος 3-opt, η μέθοδος relocate, η μέθοδος swap και η μέθοδος επανασύνδεσης διαδρομών (Path Relinking), την οποία θα αναλύσουμε λίγο παρακάτω.

3.4.3.2 Η μέθοδος 2-opt στο πρόβλημα VRP

3.4.3.2.1 Περιγραφή της μεθόδου

Η συγκεκριμένη μέθοδος αποτελείται γενικά από τη διαγραφή δυο ακμών και την επανασύνδεση δυο μονοπατιών με διαφορετικό τρόπο για να καθορίσουμε μια καινούργια διαδρομή. Παρατηρούμε ότι υπάρχει μόνο ένας τρόπος για να επανασυνδέσουμε τα μονοπάτια.

Η διαδικασία της μεθόδου για μία συγκεκριμένη διαδρομή έχει ως εξής: Πρώτα, για κάθε κόμβο i=1...,n: Εξετάζουμε 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 μπορούμε να μειώσουμε το κόστος της διαδρομής, τότε επιλέγουμε την καλύτερη 2-opt κίνηση και εφαρμόζουμε τις αλλαγές στην διαδρομή. Αυτή η διαδικασία επαναλαμβάνεται έως ότου φτάσουμε σε σημείο που δεν μπορούμε να πραγματοποιήσουμε επιπλέον βελτίωση. [1]

3.4.3.2.2 Υλοποίηση της μεθόδου 2-opt στον αλγόριθμό μας

Σε αυτό το κομμάτι του αλγορίθμου θα διαφοροποιήσουμε μία συγκεκριμένη εφικτή λύση. Καταρχήν βρίσκουμε για το αρχικό μας διάνυσμα (την αρχική λύση δηλαδή) τις ακριβότερές της διαδρομές (από κόμβο σε κόμβο). Επιλέγουμε δύο από αυτές για να τις καταργήσουμε χωρίζοντας στις αντίστοιχες θέσεις το διάνυσμα της αρχικής λύσης.

Έτσι έστω για παράδειγμα ότι έχουμε την αρχική λύση: [… i j k l m n o p …] και έχουμε επιλέξει να καταργήσουμε τις διαδρομές j→k και n→o (οι οποίες ανήκουν στις ακριβότερες διαδρομές της συγκεκριμένης λύσης). Το νέο διάνυσμα που θα πάρουμε (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: [… i j n m l k o p…]. Αυτή η λύση όπως καταλαβαίνουμε είναι σίγουρα μία εφικτή λύση. Δεν ξέρουμε όμως αν είναι καλύτερη από την αρχική, γι’ αυτό υπολογίζουμε το κόστος της. Αν το κόστος της συγκεκριμένης λύσης είναι καλύτερο ή λίγο χειρότερο από το κόστος της αρχικής τότε θα την κρατήσουμε, εάν όχι τότε επιλέγουμε μία διαδρομή από τις ακριβότερες και επιλέγουμε μία δεύτερη διαδρομή εντελώς τυχαία την οποία θα καταργήσουμε. Μετά δημιουργούμε τη νέα λύση όπως περιγράφεται παραπάνω.

Στη συνέχεια ελέγχουμε πάλι το κόστος της νέας εφικτής λύσης. Εάν το συγκεκριμένο κόστος είναι καλύτερο ή λίγο χειρότερο από αυτό της αρχικής τότε κρατάμε τη νέα λύση, εάν όχι τότε κρατάμε την προηγούμενη αρχική λύση. Μετά θα συνεχίσουμε στο επόμενο βήμα να ψάχνουμε για άλλη λύση (θέτοντας σαν αρχική λύση, τη λύση που κρατήσαμε στο προηγούμενο βήμα). Αυτή η διαδικασία επαναλαμβάνεται μέχρι να φτάσουμε σε 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.

3.4.3.3 Η μέθοδος 3-opt στο πρόβλημα VRP

3.4.3.3.1 Περιγραφή της μεθόδου

Αν θέλουμε να είμαστε πιο ευέλικτοι θα μπορούσαμε να σπάσουμε την λύση σε τρία μέρη αντί για δυο και να συνδυάσουμε τα μονοπάτια με τον καλύτερο δυνατό τρόπο.

Η διαδικασία της μεθόδου για μια συγκεκριμένη διαδρομή έχει ως εξής: Για κάθε κόμβο $i \in (V)$ υπολογίζουμε ένα σύνολο από κόμβους N (i). Για κάθε κόμβο i = 1,…,n: Εξετάζουμε τις πιθανές 3-opt κινήσεις που μπορεί να γίνουν, οι οποίες διαγράφουν από τρείς πλευρές έχοντας η κάθε μια από αυτές από μια πλευρά στην N(i). Αν με αυτό τον τρόπο μπορούμε να μειώσουμε το κόστος της διαδρομής τότε επιλέγουμε την καλύτερη 3-opt κίνηση και εφαρμόζουμε τις αλλαγές στην διαδρομή T. Αυτή η διαδικασία επαναλαμβάνεται έως ότου φτάσουμε σε σημείο που δεν μπορούμε να πραγματοποιήσουμε επιπλέον βελτίωση. [1]

3.4.3.3.2 Υλοποίηση της μεθόδου 3-opt στον αλγόριθμό μας

Και εδώ θα διαφοροποιήσουμε μία συγκεκριμένη εφικτή λύση που δίνεται σαν όρισμα εισόδου στη συνάρτηση που υλοποιεί την μέθοδο 3-opt. Βρίσκουμε για το αρχικό μας διάνυσμα (την αρχική λύση δηλαδή) τις ακριβότερές της διαδρομές (από πελάτη σε πελάτη). Επιλέγουμε τρεις από αυτές για να τις καταργήσουμε χωρίζοντας στις αντίστοιχες θέσεις το διάνυσμα της αρχικής λύσης.

Έτσι έστω για παράδειγμα ότι έχουμε την αρχική λύση: [… i j k l m n o p q r s…] και έχουμε επιλέξει να καταργήσουμε τις διαδρομές j→k, n→o, r→s (οι οποίες ανήκουν στις ακριβότερες διαδρομές της συγκεκριμένης λύσης). Το νέο διάνυσμα που θα πάρουμε (η νέα λύση δηλαδή), θα διαφέρει από την αρχική μόνο στα στοιχεία που βρίσκονται ανάμεσα στα στοιχεία j και s και θα είναι η εξής: [… i n m l k j r q p o s…]. Αυτή η λύση όπως καταλαβαίνουμε είναι σίγουρα μία εφικτή λύση. Δεν ξέρουμε όμως αν είναι καλύτερη από την αρχική, γι’ αυτό υπολογίζουμε το κόστος της.

Αν το κόστος της συγκεκριμένης λύσης είναι καλύτερο ή λίγο χειρότερο από το κόστος της αρχικής τότε θα την κρατήσουμε, εάν όχι τότε επιλέγουμε εντελώς τυχαία τις δύο από τις τρεις διαδρομές που θα καταργήσουμε και σαν τρίτη προς κατάργηση διαδρομή επιλέγουμε μία από τις ακριβότερες διαδρομές. Στη συνέχεια δημιουργούμε τη νέα λύση όπως περιγράφτηκε παραπάνω και μετά ελέγχουμε πάλι το κόστος της νέας εφικτής λύσης. Εάν το συγκεκριμένο κόστος είναι καλύτερο ή λίγο χειρότερο από αυτό της αρχικής τότε κρατάμε τη νέα λύση, εάν όχι τότε κρατάμε την προηγούμενη αρχική λύση. Μετά θα συνεχίσουμε ομοιοτρόπως στο επόμενο βήμα να ψάχνουμε για άλλη λύση (θέτοντας σαν αρχική λύση, τη λύση που κρατήσαμε στο προηγούμενο βήμα). Αυτή η διαδικασία επαναλαμβάνεται μέχρι να φτάσουμε σε ένα σημείο όπου η λύση δεν θα μπορεί να βελτιωθεί άλλο ή μέχρι να συμπληρωθεί ο μέγιστος αριθμός βημάτων της μεθόδου 3-opt. Βέβαια σε κάθε βήμα της διαδικασίας κρατάμε την καλύτερη λύση που έχει βρεθεί μέχρι εκείνη τη στιγμή, έτσι ώστε η τελική λύση που θα επιστραφεί, σαν όρισμα εξόδου από τη συνάρτηση, να είναι η καλύτερη όλων των βημάτων.

3.4.3.4 Η μέθοδος SWAP στο πρόβλημα VRP

3.4.3.4.1 Περιγραφή της μεθόδου

Με αυτήν την μέθοδο η οποία ανήκει στις μεθόδους τοπικής αναζήτησης καταργούμε τέσσερις διαδρομές από μία ήδη εφικτή λύση και τις αντικαθιστούμε με άλλες τέσσερις διαδρομές έτσι ώστε το διάνυσμα που αντιστοιχεί στη νέα εφικτή λύση να διαφέρει μόνο σε δύο στοιχεία (κόμβους) από το διάνυσμα που αντιστοιχεί στην αρχική λύση (τα οποία ανταλλάσουν μεταξύ τους θέσεις). Ουσιαστικά δύο στοιχεία (κόμβοι) αλλάζουν μεταξύ τους θέσεις. Έτσι μπορούμε να καταργήσουμε τέσσερις διαδρομές μεγάλου κόστους και να τις αντικαταστήσουμε κατάλληλα με άλλες τέσσερις οικονομικότερες διαδρομές για να βελτιωθεί το συνολικό κόστος μεταφορών. Αυτή η διαδικασία επαναλαμβάνεται έως ότου φτάσουμε σε σημείο που δεν μπορούμε να πραγματοποιήσουμε επιπλέον βελτίωση.

3.4.3.4.2 Υλοποίηση της μεθόδου SWAP στον αλγόριθμό μας

Καταρχήν ξεκινάμε από μία ήδη εφικτή λύση. Εντοπίζουμε για το αρχικό μας διάνυσμα τους κόμβους που αντιστοιχούν στις ακριβότερες διαδρομές. Οι διαδρομές που μας ενδιαφέρουν για κάποιο κόμβο είναι η διαδρομή που τον ενώνει με τον προηγούμενό του κόμβο αλλά και η διαδρομή που τον ενώνει με τον επόμενο κόμβο. Έτσι προσθέτουμε το κόστος της διαδρομής που καταλήγει στον συγκεκριμένο κόμβο με το κόστος της διαδρομής που ξεκινάει από τον ίδιο κόμβο και το τοποθετούμε σε ένα διάνυσμα που περιέχει τα κόστη για όλους τους κόμβους. Επιλέγουμε δύο από τους κόμβους που αντιστοιχούν σε ακριβότερες διαδρομές και ανταλλάζουμε τις θέσεις τους στο διάνυσμα που αναπαριστά τη λύση.

Έστω για παράδειγμα ότι έχουμε την αρχική λύση: […i j k l m n o p…] και έχουμε επιλέξει να αλλάξουμε θέσεις στους κόμβους j και n (οι οποίοι ανήκουν στους κόμβους με τις ακριβότερες διαδρομές για την συγκεκριμένη λύση). Το νέο διάνυσμα που θα πάρουμε (η νέα λύση δηλαδή), θα διαφέρει από την αρχική μόνο κατά δύο στοιχεία, τα j και n, όπου το ένα αντικατέστησε το άλλο, έτσι η νέα λύση θα έχει ως εξής: […i n k l m j o p…]. Αυτή η λύση όπως καταλαβαίνουμε είναι σίγουρα μία εφικτή λύση. Δεν ξέρουμε όμως αν είναι καλύτερη από την αρχική, γι' αυτό υπολογίζουμε το κόστος της.

Αν το κόστος της συγκεκριμένης λύσης είναι καλύτερο ή λίγο χειρότερο από το κόστος της αρχικής τότε θα την κρατήσουμε, εάν όχι τότε επιλέγουμε έναν από τους κόμβους που αντιστοιχούν σε ακριβότερες διαδρομές και έναν δεύτερο κόμβο εντελώς τυχαία. Στη συνέχεια ανταλλάζουμε θέσεις σε αυτούς τους δύο κόμβους και μετά ελέγχουμε πάλι το κόστος της νέας εφικτής λύσης. Εάν το συγκεκριμένο κόστος είναι καλύτερο ή λίγο χειρότερο από αυτό της αρχικής τότε κρατάμε τη νέα λύση, εάν όχι τότε κρατάμε την προηγούμενη αρχική λύση. Μετά θα συνεχίσουμε, με τον ίδιο τρόπο, στο επόμενο βήμα να ψάχνουμε για άλλη λύση (θέτοντας σαν αρχική λύση, τη λύση που κρατήσαμε στο προηγούμενο βήμα). Αυτή η διαδικασία επαναλαμβάνεται μέχρι να φτάσουμε σε ένα σημείο όπου η λύση δεν θα μπορεί να βελτιωθεί άλλο ή μέχρι να συμπληρωθεί ο μέγιστος αριθμός βημάτων της μεθόδου SWAP.

Όπως και στις προηγούμενες ευρετικές μεθόδους, καθ' όλη τη διάρκεια της διαδικασίας κρατάμε την καλύτερη λύση που έχει βρεθεί μέχρι στιγμής, έτσι ώστε η τελική λύση που θα επιστραφεί να μην είναι χειρότερη από την αρχική λύση.

3.4.3.5 Η μέθοδος RELOCATE στο πρόβλημα VRP

3.4.3.5.1 Περιγραφή της μεθόδου

Σε αυτήν την μέθοδο χρησιμοποιώ ένα διάνυσμα της μορφής […f i j k m n…], το οποίο αναπαριστά την αρχική λύση, την οποία θέλω να βελτιώσω. Κατά την εφαρμογή της μεθόδου απλά μεταφέρω ένα συγκεκριμένο στοιχείο (κόμβο) του διανύσματος σε άλλη θέση, μεταθέτοντας τα στοιχεία που βρίσκονται ενδιάμεσα της αρχικής και της τελικής θέσης του προαναφερθέντος στοιχείου κατά μία θέση αντίθετα προς τη φορά μεταφοράς του συγκεκριμένο στοιχείου.

3.4.3.5.2 Υλοποίηση της μεθόδου RELOCATE στον αλγόριθμό μας

Η μέθοδος RELOCATE αρχίζει από μια συγκεκριμένη εφικτή λύση. Η μέθοδος εφαρμόζεται ως εξής: Βρίσκουμε για το αρχικό μας διάνυσμα (την αρχική λύση δηλαδή) τις ακριβότερές της διαδρομές (από κόμβο σε κόμβο). Επιλέγουμε μία από αυτές για να την καταργήσουμε τοποθετώντας στην αντίστοιχη θέση έναν από τους κόμβους (στοιχεία του διανύσματος) που αντιστοιχούν σε ακριβές διαδρομές (από και προς σε αυτούς).

Έτσι αν βρω κάποιον κόμβο (στοιχείο) ο οποίος αντιστοιχεί σε ακριβές διαδρομές (από και προς σε αυτόν) και τον μεταθέσω μέσα στο διάνυσμα της λύσης, υπάρχει η ελπίδα να βελτιωθεί το συνολικό κόστος μεταφορών. Οι πιθανότητες βελτίωσης του κόστους αυξάνονται αν ο κόμβος μεταφερθεί ανάμεσα σε δύο κόμβους (έστω τους i και j) όπου η άμεση μετακίνηση από τον κόμβο i στον κόμβο j είναι πολύ ακριβή. Δηλαδή αυξάνονται οι πιθανότητες βελτίωσης του κόστους όταν καταργήσουμε μία ακριβή διαδρομή (από έναν κόμβο σε έναν άλλο).

Έτσι έστω για παράδειγμα ότι έχουμε την αρχική λύση: […f i j k m n…] και έχουμε επιλέξει να καταργήσουμε την διαδρομή i→j τοποθετώντας ανάμεσα στους κόμβους i και j τον κόμβο m (ο οποίος αντιστοιχεί σε σχετικά ακριβές διαδρομές από και προς σε αυτόν). Το νέο διάνυσμα που θα πάρουμε (η νέα λύση δηλαδή), θα διαφέρει από την αρχική στο ότι ο κόμβος m τοποθετήθηκε ανάμεσα στους κόμβους i και j και οι κόμβοι που βρίσκονταν ανάμεσα στην αρχική και την τελική θέση του κόμβου m μεταφέρθηκαν κατά μία θέση προς τα δεξιά (προς την αντίθετη δηλαδή κατεύθυνση από αυτήν της μετακίνησης του κόμβου m). Δηλαδή η νέα λύση θα είναι η εξής: […f i m j k n…].

Αυτή η λύση όπως καταλαβαίνουμε είναι σίγουρα μία εφικτή λύση. Δεν ξέρουμε όμως αν είναι καλύτερη από την αρχική, γι’ αυτό υπολογίζουμε το κόστος της. Αν το κόστος της συγκεκριμένης λύσης είναι καλύτερο ή λίγο χειρότερο από το κόστος της αρχικής τότε θα την κρατήσουμε, εάν όχι τότε επιλέγουμε μία διαδρομή εντελώς τυχαία την οποία θα καταργήσουμε τοποθετώντας στην αντίστοιχη θέση έναν από τους κόμβους (στοιχεία του διανύσματος) που αντιστοιχούν σε σχετικά ακριβές διαδρομές (από και προς σε αυτούς). Μετά δημιουργούμε τη νέα λύση όπως περιγράφτηκε παραπάνω.

Στη συνέχεια ελέγχουμε πάλι το κόστος της νέας εφικτής λύσης. Εάν το συγκεκριμένο κόστος είναι καλύτερο ή λίγο χειρότερο από αυτό της αρχικής τότε κρατάμε τη νέα λύση, εάν όχι τότε κρατάμε την προηγούμενη αρχική λύση. Μετά θα συνεχίσουμε στο επόμενο βήμα να ψάχνουμε για άλλη λύση (θέτοντας σαν αρχική λύση, τη λύση που κρατήσαμε στο προηγούμενο βήμα). Αυτή η διαδικασία επαναλαμβάνεται μέχρι να φτάσουμε σε ένα σημείο όπου η λύση δεν θα μπορεί να βελτιωθεί άλλο ή μέχρι να συμπληρωθεί ο μέγιστος αριθμός βημάτων της μεθόδου RELOCATE. Τέλος επιστρέφεται σαν αποτέλεσμα η καλύτερη λύση που προέκυψε από όλα τα βήματα της διαδικασίας.

Με αυτόν λοιπόν τον τρόπο βελτιώνουμε τις αρχικές μας λύσεις. Αυτές λοιπόν τις αρχικές λύσεις τις χωρίζουμε σε επιμέρους ομάδες (νησιά), τα οποία θα εξελίσσονται παράλληλα. Περιστασιακά θα γίνεται “μετανάστευση” των καλύτερων λύσεων από ομάδα σε ομάδα, μια διαδικασία που η υλοποίησή της θα περιγραφεί αναλυτικά αργότερα.

Στη συνέχεια εργαζόμαστε ξεχωριστά σε κάθε μία ομάδα (νησί) και ταξινομούμε τις λύσεις της από την καλύτερη προς την χειρότερη. Αργότερα για κάθε ομάδα ξεχωριστά εργαζόμαστε ως εξής: Παίρνουμε ανά δύο γειτονικές λύσεις και τις διασταυρώνουμε (π.χ. την πρώτη καλύτερη με την δεύτερη καλύτερη). Αυτή η διαδικασία επαναλαμβάνεται για όλες τις λύσεις της ομάδας που εργαζόμαστε.

Η διασταύρωση γίνεται δίνοντας σαν βασικά ορίσματα εισόδου δύο λύσεις που πρόκειται να διασταυρωθούν και πραγματοποιείται όπως περιγράφεται στην παρακάτω σελίδα:

3.4.3.6 Υλοποίηση Διασταύρωσης

Για να υλοποιηθεί η διασταύρωση χρειάζονται δύο αρχικές λύσεις. Βρίσκουμε για κάθε διάνυσμα (αρχική λύση) τις ακριβότερες της διαδρομές (από πελάτη σε πελάτη). Επιλέγουμε μία από αυτές για να την καταργήσουμε χωρίζοντας στην αντίστοιχη θέση τα διανύσματα των αρχικών λύσεων και δίνουμε στον απόγονο το πρώτο κομμάτι από το ένα διάνυσμα και το δεύτερο κομμάτι από το άλλο διάνυσμα.

Το νέο διάνυσμα, το οποίο αντιστοιχεί στον απόγονο, κατά πάσα πιθανότητα είναι μία μη εφικτή λύση. Γι' αυτό το λόγο μέσω της διαδικασίας της μετάλλαξης (η οποία θα περιγραφεί αργότερα) αυτό το διάνυσμα θα αλλάξει και θα γίνει μία εφικτή λύση. Θα εφαρμόσουμε επιπλέον τοπική αναζήτηση στην συγκεκριμένη λύση με τέσσερις τρόπους (2-opt, 3-opt, swap, relocate). Η τοπική αναζήτηση γίνεται σε ξεχωριστά m-file για κάθε έναν τρόπο.

Στο τέλος αν η τελική λύση που προκύπτει μετά και από την τοπική αναζήτηση είναι καλύτερη ή λίγο χειρότερη από το ($1^η$) αρχική λύση, θα την κρατήσουμε για τη συνέχεια της διαδικασίας, αλλιώς δεν την κρατάμε. Αντί αυτής θα κρατήσουμε μία λύση προερχόμενη από τη μέθοδο της Άπληστης Τυχοποιημένης Προσαρμοστικής Αναζήτησης (Greedy Randomized Adaptive Search Procedure – GRASP) που περιγράφτηκε προηγουμένως.

Η διαδικασία της μετάλλαξης που αναφέρθηκε προηγούμενα πραγματοποιείται όπως περιγράφεται στην παρακάτω σελίδα:

3.4.3.7 Μετάλλαξη

Καταρχήν εντοπίζουμε τους κόμβους που δεν βρίσκονται στη μη εφικτή λύση την οποία προσπαθούνε να διορθώσουμε - “μεταλλάξουμε”. Αυτό γίνεται επειδή σύμφωνα με την μοντελοποίηση που ακολουθήθηκε στο πρόβλημα, μία εφικτή λύση περιέχει όλους τους κόμβους του προβλήματος από μία φορά τον καθένα. Όλοι λοιπόν οι κόμβοι που δεν υπάρχουν στην μη εφικτή λύση που εξετάζουμε αποθηκεύονται σε ένα διάνυσμα (έστω το y).

Στη συνέχεια θα εντοπίσουμε σε ποιες θέσεις του διανύσματος υπάρχουν κόμβοι οι οποίοι δεν είναι μοναδικοί στο συγκεκριμένο διάνυσμα της λύσης. Στη συνέχεια οι επιπλέον θέσεις αποθηκεύονται σε ένα δεύτερο διάνυσμα (έστω το z). Αν παραδείγματος χάριν εντοπίσουμε έναν συγκεκριμένο κόμβο σε τρεις θέσεις στο διάνυσμα της λύσης (έστω στην i-οστή θέση, στην j-οστή θέση και στην x-οστή θέση) τότε θα αποθηκεύσουμε στο διάνυσμα z τους αριθμούς j και x (θα μπορούσαμε να αποθηκεύσουμε τους αριθμούς i και j ή τους αριθμούς i και x). Αργότερα σε όλες τις θέσεις των οποίων ο αριθμός υπάρχει στο διάνυσμα z θα τοποθετήσουμε κόμβους οι οποίοι δεν υπάρχουν στην αρχική μη εφικτή λύση. Γι αυτό το λόγο τοποθετούμε στο διάνυσμα z όλες τις θέσεις (ενός πολλαπλού κόμβου) εκτός μίας, επειδή κάθε κόμβος θα πρέπει να υπάρχει μία φορά στο τελικό εφικτό διάνυσμα (εκτός του κόμβου 1 φυσικά που δεν υπάρχει καμία φορά σε ένα εφικτό διάνυσμα).

Όταν λοιπόν τελειώσει η συμπλήρωση του διανύσματος z, αρχίζουμε να αλλάζουμε τα στοιχεία της αρχικής μη εφικτής λύσης στις θέσεις που υποδεικνύει το διάνυσμα z. Κάθε φορά αντικαθιστούμε ένα στοιχείο της αρχικής μη εφικτής λύσης με ένα στοιχείο από το διάνυσμα y το οποίο περιέχει τους κόμβους που δεν περιείχε η αρχική εφικτή λύση. Κάθε φορά που εισάγεται ένας κόμβος από το διάνυσμα y στο διάνυσμα της αρχικής μη εφικτής λύσης, διαγράφεται από το διάνυσμα y έτσι ώστε να μην συμπληρωθεί για δεύτερη φορά ο ίδιος κόμβος στη λύση που προσπαθούμε να κάνουμε εφικτή.

Αξίζει να σημειωθεί πως σε κάθε αντικατάσταση κόμβου, επιλέγεται από το διάνυσμα y ένας κόμβος από τους πιο κοντινούς, στους γειτονικούς κόμβους της θέσης του διανύσματος που πρόκειται να γίνει η αλλαγή. Δηλαδή αν πρόκειται να αντικατασταθεί ο κόμβος που βρίσκεται στη θέση i του διανύσματος της λύσης, εμείς θα εντοπίσουμε ποιοι από τους κόμβους του διανύσματος y είναι πιο κοντινοί στους κόμβους που βρίσκονται στην i-1 ή και i+1 θέση του διανύσματος της λύσης. Στη συνέχεια θα επιλεχθεί ένας από αυτούς τους κόμβους για να εισαχθεί στη θέση i του διανύσματος της λύσης.

Έτσι στο τέλος της διαδικασίας της μετάλλαξης, παίρνουμε μία όχι μόνο εφικτή αλλά και σχετικά καλή λύση.

Στην συνέχεια ξανα-ταξινομούμε τις λύσεις της ομάδας όπου εργαζόμαστε με βάση το κόστος τους, ξεκινώντας από την καλύτερη και καταλήγοντας στην χειρότερη. Μετά θα εφαρμόσουμε τη μέθοδο Path Relinking μεταξύ των δύο πρώτων λύσεων της ομάδας, δηλαδή μεταξύ των δύο καλύτερων λύσεων της ομάδας.

Η μέθοδος Path Relinking πραγματοποιείται όπως περιγράφεται στην επόμενη σελίδα:

3.4.3.8 Η μέθοδος PATH RELINKING στο πρόβλημα VRP

3.4.3.8.1 Περιγραφή της μεθόδου

Η μέθοδος PATH RELINKING είναι μια έξυπνη μέθοδος τοπικής αναζήτησης. Σύμφωνα με αυτήν την μέθοδο, ξεκινάμε από δύο εφικτές λύσεις. Έστω ότι κάθε λύση αναπαριστάται από ένα διάνυσμα. Σε κάθε βήμα αλλάζουμε ένα στοιχείο της πρώτης λύσης και το κάνουμε ίδιο με το αντίστοιχο στοιχείο της δεύτερης λύσης υπολογίζοντας το κόστος της ενδιάμεσης λύσης που προέκυψε. Αυτή η διαδικασία επαναλαμβάνεται έως ότου η πρώτη λύση γίνει ίση με τη δεύτερη λύση. Για την εύρεση της τελικής λύσης θα πρέπει να συγκρίνουμε τα κόστη όλων των ενδιάμεσων αλλά και των δύο αρχικών λύσεων. Μετά τη σύγκριση θα κρατήσουμε τη λύση με το μικρότερο κόστος.

3.4.3.8.2 Υλοποίηση της μεθόδου PATH RELINKING στον αλγόριθμό μας

Η μέθοδος PATH RELINKING στον αλγόριθμό μας εφαρμόζεται λαμβάνοντας ως αρχικές δύο ήδη σχετικά καλές λύσεις. Έτσι υπάρχει αυξημένη πιθανότητα από δύο ήδη καλές λύσεις να οδηγηθούμε σε μία καλύτερη.

Αρχικά αποθηκεύουμε τη πρώτη αρχική λύση στην πρώτη γραμμή ενός πίνακα, ο οποίος συμπεριλαμβάνει όλες τις λύσεις. Επίσης αποθηκεύουμε το κόστος της ίδιας λύσης στην πρώτη θέση ενός διανύσματος που περιέχει τα κόστη. Μετά τοποθετούμε την 1^{η} αρχική λύση και στην δεύτερη γραμμή αλλάζοντάς της το πρώτο στοιχείο της και κάνοντάς το ίδιο με αυτό της δεύτερης αρχικής λύσης. Καθιστούμε εφικτή τη νέα λύση τοποθετώντας το στοιχείο που αφαιρέσαμε (από την πρώτη θέση), στη θέση που υπήρχε πριν το στοιχείο που προσθέσαμε (στην πρώτη θέση). Στη συνέχεια αποθηκεύουμε το κόστος της λύσης αυτής, στην δεύτερη θέση του διανύσματος που περιέχει τα κόστη. Μετά τοποθετούμε τη λύση της δεύτερης γραμμής και στην τρίτη γραμμή αλλάζοντάς της μόνο το δεύτερο στοιχείο της και κάνοντάς το ίδιο με αυτό της δεύτερης αρχικής λύσης. Και ομοίως με πριν καθιστούμε εφικτή τη νέα λύση και αποθηκεύουμε το κόστος της στην τρίτη θέση του διανύσματος με τα κόστη. Αυτή η διαδικασία επαναλαμβάνεται έως ότου η λύση που τροποποιείται να γίνει ίδια με την δεύτερη αρχική λύση.

Στο τέλος συγκρίνουμε τα κόστη όλων των λύσεων που προέκυψαν σε όλα τα βήματα της διαδικασίας. Μετά επιλέγουμε δύο από τις καλύτερες λύσεις, οι οποίες πρέπει να είναι διαφορετικές μεταξύ τους, και τις δίνουμε σαν όρισμα εξόδου από τη συνάρτηση για να χρησιμοποιηθούν στη συνέχεια από το κύριο πρόγραμμα. Εάν οι δύο αυτές λύσεις είναι καλύτερες από τις αρχικές, θα αντικαταστήσουν τις χειρότερες λύσεις της ομάδας και όχι τις αρχικές που χρησιμοποιήθηκαν σαν όρισμα εισόδου από τη συνάρτηση της μεθόδου Path Relinking. Αυτό γίνεται για να μην χαθούν δύο πολύ καλές λύσεις, είναι προτιμότερο να χαθούν δύο σχετικά χειρότερες λύσεις.

Στην συνέχεια ταξινομούνται και πάλι οι λύσεις της ομάδας, από την καλύτερη έως τη χειρότερη. Με αυτόν τον τρόπο αν υπάρχουν ίδιες ή παρόμοιες λύσεις στην ομάδα, θα γίνουν γειτονικές ή έστω θα έρθουν πολύ κοντά μετά την ταξινόμηση. Έτσι ελέγχω αν όλα τα πιθανά ζευγάρια γειτονικών και πολύ κοντινών λύσεων έχουν μεγάλες ομοιότητες. Εάν βρεθούν λύσεις που μοιάζουν πάρα πολύ, τότε διαγράφονται οι χειρότερες. Αυτή είναι μία πολύ σημαντική διαδικασία η οποία βοηθάει καθοριστικά στην επιβράδυνση της σύγκλισης του αλγορίθμου η οποία με τη σειρά της εγγυάται ένα πολύ καλύτερο αποτέλεσμα. Εάν ο αριθμός των λύσεων της ομάδας μειωθεί πάρα πολύ τότε συμπληρώνω με λύσεις από τη μέθοδο της Άπληστης Τυχοποιημένης Προσαρμοστικής Αναζήτησης (Greedy Randomized Adaptive Search Procedure – GRASP) που περιγράφτηκε προηγούμενα. Αυτή ακριβώς η διαδικασία συνεχίζεται και στα επόμενα βήματα – γενιές, έως ότου συμπληρωθεί ο μέγιστος αριθμός γενεών. Αξίζει να σημειωθεί ότι εάν, για έναν ικανοποιητικό αριθμό βημάτων, δεν αλλάξουν οι καλύτερες λύσεις της ομάδας, τότε στα επόμενα βήματα δεν θα διασταυρώνονται πια γειτονικές λύσεις (δηλαδή οι καλές με τις καλές και οι μέτριες με τις μέτριες), αλλά με τυχαίο τρόπο. Με τυχαίο τρόπο θα συνεχίσουν να διασταυρώνονται έως ότου αλλάξουν οι καλύτερες λύσεις της ομάδας.

Αυτή η διαδικασία επαναλαμβάνεται για κάθε μία ομάδα. Αφού λοιπόν η παραπάνω διαδικασία ολοκληρωθεί για όλες τις ομάδες, θα αρχίσει η διαδικασία της μετανάστευσης.

Κατά τη μετανάστευση οι καλύτερες λύσεις μίας ομάδας – “νησιού” αντικαθιστούν τις χειρότερες λύσεις της επόμενης ομάδας. Αυτή η διαδικασία επαναλαμβάνεται για όλα τα ζευγάρια των γειτονικών ομάδων – “νησιών”. Ακολουθείται δηλαδή τοπολογία δακτυλίου, όπου οι μετανάστες - λύσεις μεταναστεύουν σε διπλανό νησί - ομάδα και αντικαθιστούν ένα μέρος του πληθυσμού της συγκεκριμένης ομάδας.

Αυτή η διαδικασία συνεχίζεται έως ότου συμπληρωθεί ο μέγιστος αριθμός μεταναστεύσεων ή έως ότου όλα τα νησιά αποκτήσουν την ίδια λύση σαν καλύτερή τους, σταματάμε τις μεταναστεύσεις και υπάρχει η δυνατότητα να συνεχίσουμε την ίδια διαδικασία πάνω σε μία όμως μονάχα ομάδα και να σταματήσουμε την διαδικασία μετά από ένα συγκεκριμένο αριθμό βημάτων-γενιών. Αλλιώς σταματάμε τώρα τη διαδικασία.

Τέλος, επιλέγουμε την καλύτερη λύση η οποία θα υποστεί περεταίρω τοπική αναζήτηση με τις μεθόδους 2-opt, 3-opt, swap και relocate. Έτσι προκύπτει η τελική λύση την οποία θα σχεδιάσουμε.

Ο σχεδιασμός της βέλτιστης λύσης γίνεται παίρνοντας ως βασικά δεδομένα τις συντεταγμένες του κάθε κόμβου και το διάνυσμα της βέλτιστης λύσης. Έτσι τοποθετούμε τους κόμβους στο σχήμα σύμφωνα με τις συντεταγμένες τους. Επίσης αφού η λύση περιέχεται σε ένα μονάχα διάνυσμα θα πρέπει να πάρουμε υπόψη μας τους χρόνους εξυπηρέτησης, τους χρόνους ταξιδίου μεταξύ των κόμβων καθώς και τη ζήτηση του κάθε κόμβου. Έτσι όταν σταματήσει να ικανοποιείται κάποιος περιορισμός (είτε ο χρονικός περιορισμός είτε αυτός που αναφέρεται στην χωρητικότητα των φορτηγών) τότε το φορτηγό επιστρέφει στον αρχικό κόμβο – αποθήκη και στη συνέχεια μία νέα διαδρομή ξεκινάει από τον αρχικό κόμβο προς τον επόμενο. Έτσι σχηματίζουμε όλες τις διαδρομές οι οποίες ξεκινούν και καταλήγουν στον αρχικό κόμβο – αποθήκη. Τέλος η λύση αποθηκεύεται σε μορφή διανύσματος σε ένα αρχείο excel.

3.4.3.9 Παραλλαγές που υλοποιήθηκαν εκτός του Μεμετικού Αλγορίθμου Πολλαπλών Πληθυσμών – Νησιών

Στο επόμενο κεφάλαιο που περιέχει τα αποτελέσματα του Μεμετικού Αλγορίθμου Πολλαπλών Πληθυσμών – Νησιών , για να φανεί η αποτελεσματικότητα του συγκεκριμένου αλγορίθμου, παρουσιάζονται και τα αποτελέσματα από άλλες τρεις παραλλαγές αυτού του αλγορίθμου που υλοποιήθηκαν επίσης στην συγκεκριμένη εργασία. Οι τρεις παραλλαγές έχουν ως εξής:

Στην πρώτη παραλλαγή εφαρμόζουμε ακριβώς τον ίδιο αλγόριθμο που περιγράφτηκε πριν με την εξής όμως διαφορά: Στον βασικό αλγόριθμο, για την εύρεση μιας σχετικά καλής (αρχικής) λύσης χρησιμοποιείται η μέθοδος της Άπληστης Τυχοποιημένης Προσαρμοστικής Αναζήτησης (GRASP). Αντίθετα στην πρώτη παραλλαγή, στα αντίστοιχα σημεία του αλγορίθμου παράγουμε τις λύσεις με εντελώς τυχαίο τρόπο. Οι παράμετροι αυτής της παραλλαγής έχουν ως εξής: ο αριθμός των νησιών – ομάδων είναι 5, ο πληθυσμός του κάθε νησιού είναι 20, η μετανάστευση γίνεται κάθε 800 γενιές και τα βήματα του αλγορίθμου είναι 13. Επίσης ισχύει τόσο για αυτήν την παραλλαγή όσο και για το βασικό αλγόριθμο (όπου κάναμε χρήση της μεθόδου “GRASP”) ότι ο μέγιστος αριθμός επαναλήψεων local search είναι 100 για κάθε μέθοδο τοπικής αναζήτησης που χρησιμοποιούμε και 400 (4×100) για κάθε βήμα μέσα σε μία γενιά. Άρα με δεδομένο ότι εκτελούμε (σε όλες τις παραλλαγές) 20 βήματα σε κάθε γενιά, ο μέγιστος αριθμός επαναλήψεων local search για κάθε βήμα του αλγορίθμου είναι $5 \times 800 \times 20 \times 400 = 32.000.000$.

Η δεύτερη παραλλαγή, έχει την εξής διαφορά από το βασικό αλγόριθμο: Στην συγκεκριμένη παραλλαγή υλοποιείται απλά ο Μεμετικός Αλγόριθμος. Δεν γίνεται χρήση διάφορων ομάδων (νησιών) λύσεων και συνεπώς δεν πραγματοποιείται ούτε μετανάστευση. Αντίθετα εργαζόμαστε πάνω σε μία μονάχα ομάδα λύσεων. Οι παράμετροι αυτής της παραλλαγής έχουν ως εξής: ο αριθμός των γενεών είναι ίσος με 20.000 και ο μέγιστος αριθμός επαναλήψεων local search σε κάθε γενιά είναι $400 \times 20 = 8.000$. Αφού και στις δύο τελευταίες παραλλαγές ο μέγιστος αριθμός επαναλήψεων local search είναι 400 (4×100) για κάθε βήμα μέσα σε μία γενιά.

Στην τρίτη παραλλαγή υλοποιείται απλά ο Μεμετικός Αλγόριθμος χωρίς τη χρήση της μεθόδου της Άπληστης Τυχοποιημένης Προσαρμοστικής Αναζήτησης (GRASP). Στον βασικό αλγόριθμο για την εύρεση μιας σχετικά καλής (αρχικής) λύσης. χρησιμοποιήθηκε η μέθοδος GRASP. Αντίθετα στα αντίστοιχα σημεία του αλγορίθμου της τρίτης παραλλαγής οι λύσεις παράγονται εντελώς τυχαία. Οι παράμετροι αυτής της παραλλαγής έχουν ως εξής ο αριθμός των γενεών είναι ίσος με 20.000 και ο μέγιστος αριθμός επαναλήψεων local search σε κάθε γενιά είναι $400 \times 20 = 8.000$.

ΚΕΦΑΛΑΙΟ 4

Περιγραφή των προβλημάτων που επιλύθηκαν - Παρουσίαση των αποτελεσμάτων

4.1 Εισαγωγή

Στα παραδείγματα που επιλύσαμε χρησιμοποιήσαμε τις παραμέτρους που παρουσιάζονται στον παρακάτω πίνακα :

ΑΡΙΘΜΟΣ ΝΗΣΩΝ	ΠΛΗΘΥΣΜΟΣ ΚΑΘΕ ΝΗΣΟΥ	ΣΥΧΝΟΤΗΤΑ ΜΕΤΑΝΑΣΤΕΥΣΗΣ	ΒΗΜΑΤΑ ΑΛΓΟΡΙΘΜΟΥ	ΜΕΓ. ΑΡΙΘΜΟΣ ΕΠΑΝΑΛΗΨΕΩΝ LOCAL SEARCH
5	20	Κάθε 800 γενιές	13	100

Άρα ο μέγιστος αριθμός επαναλήψεων local search ανά βήμα του Μεμετικού Αλγορίθμου Πολλαπλών Πληθυσμών-Νησιών είναι $5 \times 20 \times 800 \times 4 \times 100 = 32.000.000$. Συγκεντρωτικά, τα αποτελέσματά μας για τον συγκεκριμένο αλγόριθμο στο πρόβλημα VRP, φαίνονται στον παρακάτω πίνακα, (αναλυτικές πληροφορίες για κάθε ένα παράδειγμα δίνονται στο παράρτημα):

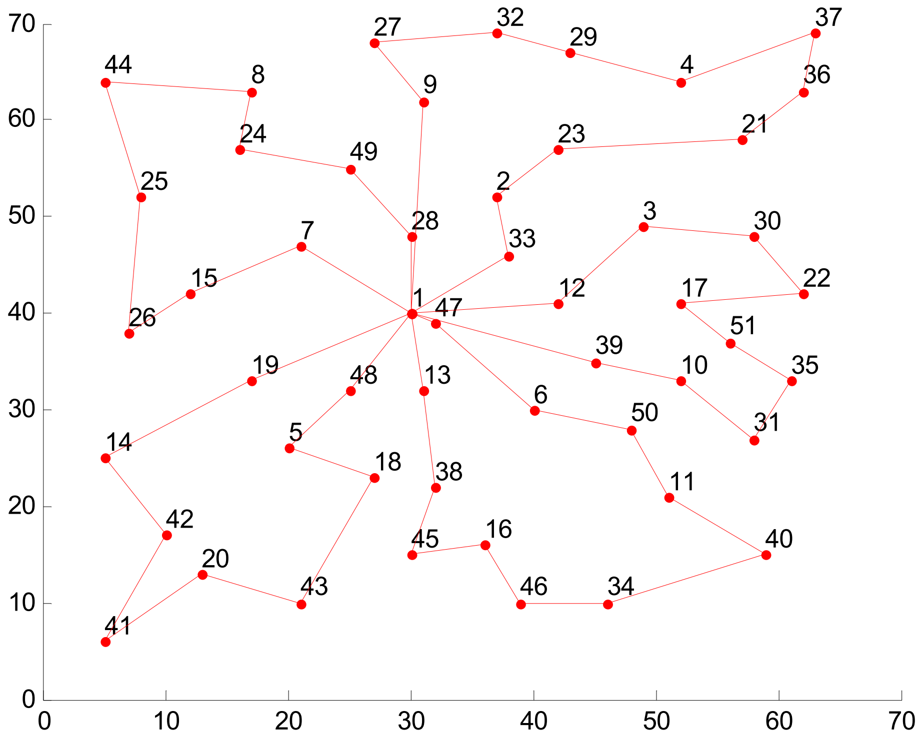
ΠΑΡΑΔΕΙΓΜΑ	ΚΟΜΒΟΙ	ΧΩΡΗΤΙΚΟΤΗΤΑ ΦΟΡΤΗΓΩΝ	ΠΕΡΙΟΡΙΣΜΟΣ ΔΙΑΔΡΟΜΗΣ	ΜΕΧΡΙ ΣΤΙΓΜΗΣ ΒΕΛΤΙΣΤΟ	ΑΠΟΤΕΛΕΣΜΑ ΑΛΓΟΡΙΘΜΟΥ ΜΑΣ	ΑΠΟΚΛΙΣΗ(%) ΑΠΟ ΤΟ ΜΕΧΡΙ ΣΤΙΓΜΗΣ ΒΕΛΤΙΣΤΟ
1	51	160	-	524,61	**524,6111**	**0%**
2	76	140	-	835,26	839,3487	0,49%
3	101	200	-	826,14	852,0737	3,14%
4	151	200	-	1028,42	1072,8858	4,32%
5	200	200	-	1291,29	1387,9302	7,48%
6	51	160	200	555,43	**555,4302**	**0%**
7	76	140	160	909,68	926,9197	1,90%
8	101	200	230	865,94	902,9253	4,10%
9	151	200	200	1162,55	1333,8350	14,73%
10	200	200	200	1395,85	1553,3715	11,28%
11	101	200	-	819,56	1048,1585	0,58%
12	121	200	-	1042,11	858,6131	4,77%
13	101	200	720	866,37	1563,8016	1,47%
14	121	200	1040	1541,14	876,8419	1,21%

Παρατηρούμε ότι τα αποτελέσματα είναι πολύ ικανοποιητικά. Μάλιστα στα παραδείγματα 1 και 6 βρήκαμε αποτελέσματα ίσα με τα αντίστοιχα μέχρι στιγμής βέλτιστα που έχουν βρεθεί σε παγκόσμια κλίμακα.

4.2 Παράδειγμα 1

Σε αυτό το παράδειγμα υπάρχουν 51 κόμβοι. Ο κόμβος 1 αντιστοιχεί στην αποθήκη και οι υπόλοιποι 50 στους πελάτες. Κάθε πελάτης έχει συγκεκριμένες συντεταγμένες θέσης και συγκεκριμένη ζήτηση, ακριβή νούμερα για τα οποία δίνονται στο παράρτημα. Θεωρείται ότι όλα τα διαθέσιμα φορτηγά έχουν την ίδια χωρητικότητα, η οποία είναι ίση με 160. Θεωρείται ότι δεν υπάρχουν χρονικοί περιορισμοί.

Για το συγκεκριμένο παράδειγμα, τα αποτελέσματα και των τεσσάρων αλγορίθμων που υλοποιήθηκαν στα πλαίσια αυτής της διπλωματικής εργασίας ήταν ίδια μεταξύ τους. Ίδια ήταν επίσης η λύση μας με το μέχρις στιγμής βέλτιστο αποτέλεσμα που έχει βρεθεί από ανάλογες έρευνες σε παγκόσμια κλίμακα για το συγκεκριμένο παράδειγμα. Η βέλτιστη λύση λοιπόν που βρήκαμε έχει κόστος ίσο με 524,6111 και γραφικά έχει ως εξής:



Η συγκεκριμένη λύση, όπως φαίνεται και από το σχήμα, αποτελείται από πέντε διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι πέντε αυτές διαδρομές έχουν ως εξής:

1	28	49	24	8	44	25	26	15	7	1		
1	33	2	23	21	36	37	4	29	32	27	9	1
1	39	10	31	35	51	17	22	30	3	12	1	
1	19	14	42	41	20	43	18	5	48	1		
1	13	38	45	16	46	34	40	11	50	6	47	1

4.3 Παράδειγμα 2

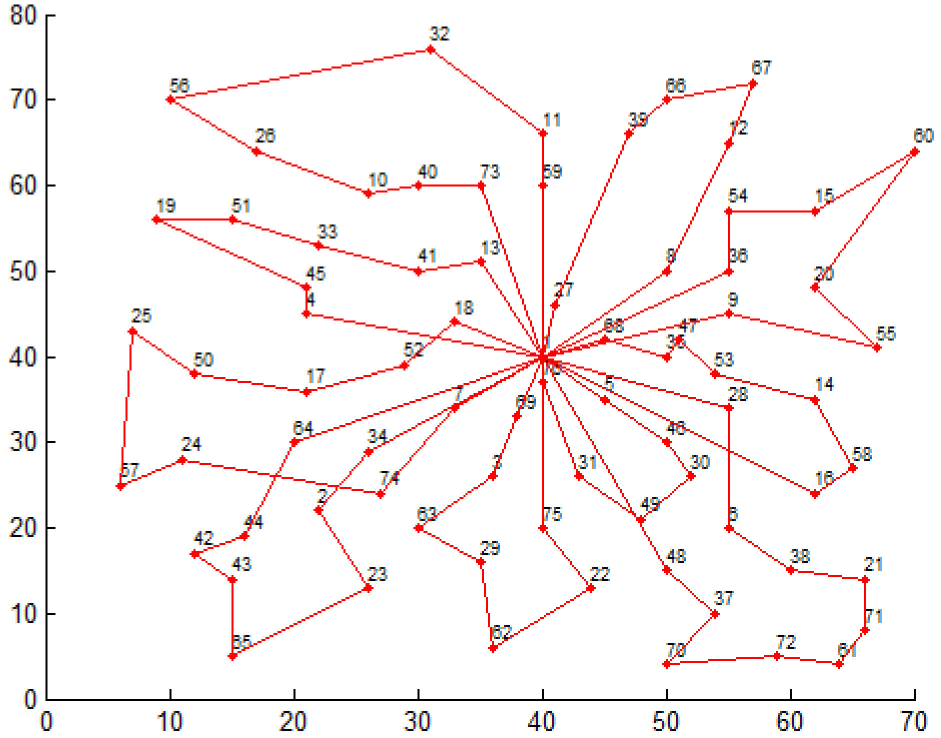
Στο δεύτερο παράδειγμα υπάρχουν 76 κόμβοι όπου ο κόμβος 1 αντιστοιχεί στην αποθήκη και οι υπόλοιποι 75 στους πελάτες. Κάθε πελάτης έχει συγκεκριμένες συντεταγμένες θέσης και συγκεκριμένη ζήτηση. Οι τιμές για τις συντεταγμένες των κόμβων και τη ζήτησή τους δίνονται στο παράρτημα. Επίσης θεωρείται ότι όλα τα διαθέσιμα φορτηγά έχουν την ίδια χωρητικότητα, η οποία είναι ίση με 140. Τέλος δεν υπάρχουν χρονικοί περιορισμοί.

<u>Αποτελέσματα για τον απλό Μεμετικό Αλγόριθμο</u>

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο έχει κόστος 862,3719 . Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (835,26) είναι:

$$\frac{862{,}3719 - 835{,}26}{835{,}26} = 3{,}25\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από δέκα διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι δέκα αυτές διαδρομές έχουν ως εξής:

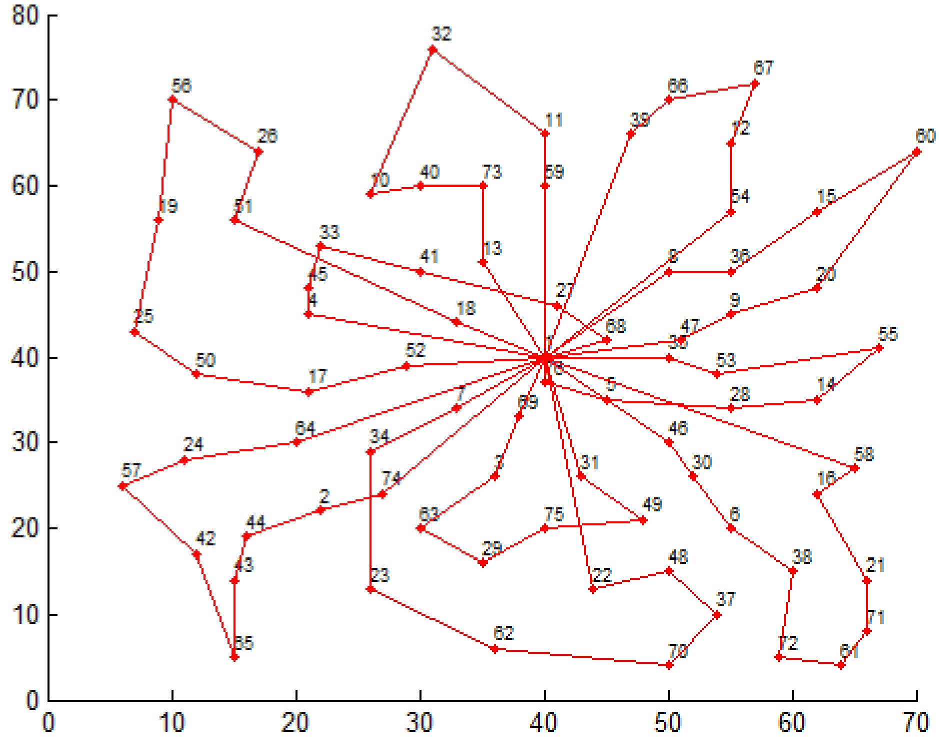
1	8	12	67	66	39	27	1				
1	13	41	33	51	19	45	4	1			
1	64	44	42	43	65	23	2	34	1		
1	5	46	30	49	31	76	1				
1	7	74	24	57	25	50	17	52	18	1	
1	59	11	32	56	26	10	40	73	1		
1	68	35	47	53	14	58	16	1			
1	28	6	38	21	71	61	72	70	37	48	1
1	69	3	63	29	62	22	75	1			
1	36	54	15	60	20	55	9	1			

Αποτελέσματα για τον Μεμετικό Αλγόριθμο με χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο όπου έγινε χρήση της μεθόδου GRASP έχει κόστος 856.7914. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (835,26) είναι:

$$\frac{856{,}7914 - 835{,}26}{835{,}26} = 2{,}58\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από δέκα διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι δέκα αυτές διαδρομές έχουν ως εξής:

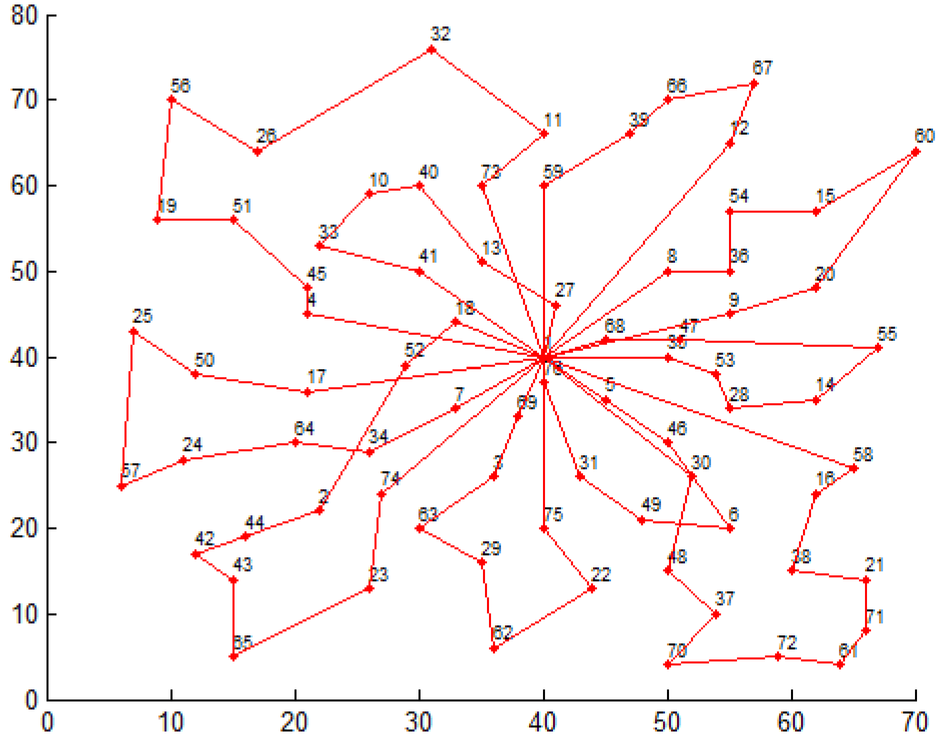
1	47	9	20	60	15	36	8	1			
1	54	12	67	66	39	1					
1	58	16	21	71	61	72	38	6	30	46	1
1	31	49	75	29	63	3	69	1			
1	22	48	37	70	62	23	34	7	1		
1	74	2	44	43	65	42	57	24	64	1	
1	76	5	28	14	55	53	35	1			
1	18	51	26	56	19	25	50	17	52	1	
1	4	45	33	41	27	68	1				
1	13	73	40	10	32	11	59	1			

Αποτελέσματα για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών, μοιάζει πολύ με την λύση της προηγούμενης παραλλαγής του αλγορίθμου και είναι ελαφρώς βελτιωμένη. Έχει κόστος 847,2109. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (835,26) είναι:

$$\frac{847{,}2109 - 835{,}26}{835{,}26} = 1{,}43\%$$

Γραφικά η λύση έχει ως εξής:



Η συγκεκριμένη λύση, όπως φαίνεται και από το σχήμα, αποτελείται επίσης από δέκα διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι δέκα αυτές διαδρομές έχουν ως εξής:

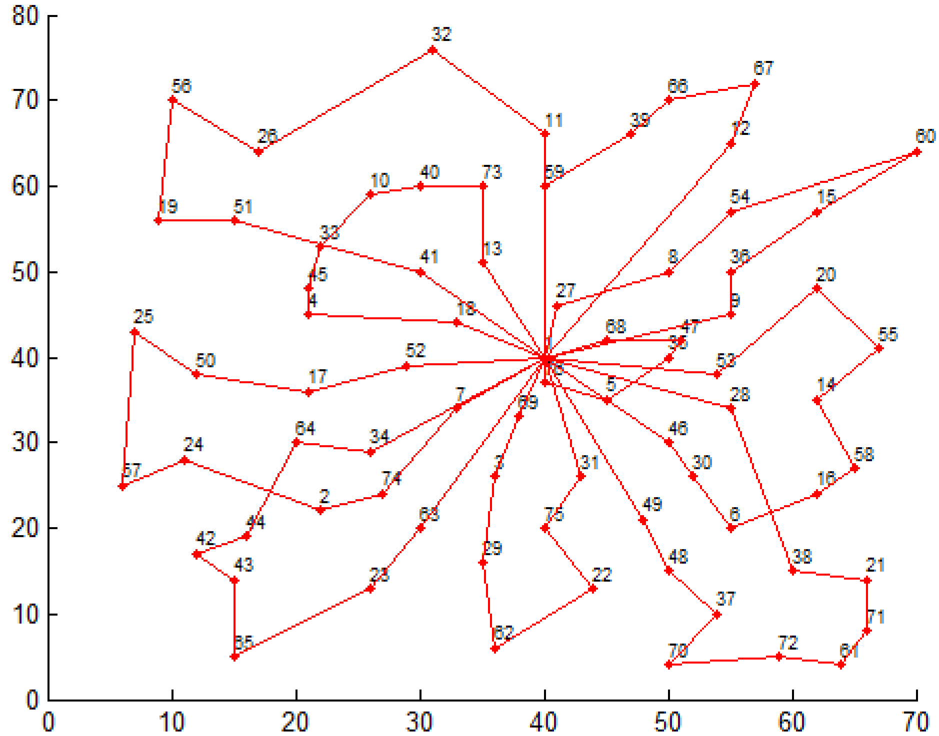
1	41	33	10	40	13	27	1					
1	7	34	64	24	57	25	50	17	1			
1	74	23	65	43	42	44	2	52	18	1		
1	69	3	63	29	62	22	75	1				
1	8	36	54	15	60	20	9	1				
1	35	53	28	14	55	47	68	1				
1	58	16	38	21	71	61	72	70	37	48	30	1
1	5	46	6	49	31	76	1					
1	4	45	51	19	56	26	32	11	73	1		
1	12	67	66	39	59	1						

Αποτελέσματα για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών με χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών όπου έγινε χρήση της μεθόδου GRASP έχει κόστος 839,3487. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (835,26) είναι μόλις:

$$\frac{839{,}3487 - 835{,}26}{835{,}26} = 0{,}49\%$$

Γραφικά έχει ως εξής:



Αυτή η λύση αποτελείται από δέκα διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι δέκα αυτές διαδρομές έχουν ως εξής:

1	27	8	54	60	15	36	9	1			
1	46	30	6	16	58	14	55	20	53	1	
1	52	17	50	25	57	24	2	74	7	1	
1	41	51	19	56	26	32	11	1			
1	63	23	65	43	42	44	64	34	1		
1	13	73	40	10	33	45	4	18	1		
1	76	5	35	47	68	1					
1	59	39	66	67	12	1					
1	31	75	22	62	29	3	69	1			
1	28	38	21	71	61	72	70	37	48	49	1

4.4 Παράδειγμα 3

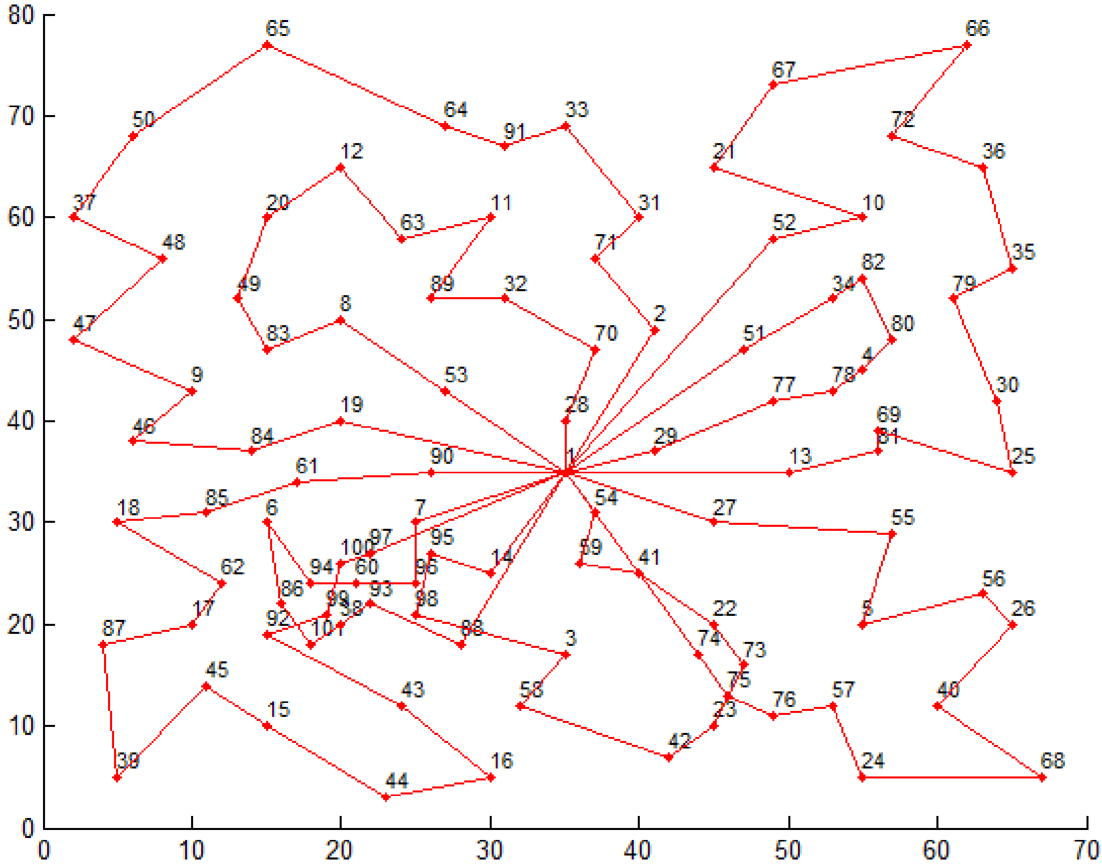
Σε αυτό το παράδειγμα υπάρχουν 101 κόμβοι. Ο κόμβος 1 αντιστοιχεί στην αποθήκη και οι υπόλοιποι 100 στους πελάτες. Κάθε πελάτης έχει συγκεκριμένες συντεταγμένες θέσης και συγκεκριμένη ζήτηση, ακριβή νούμερα για τα οποία δίνονται στο παράρτημα. Θεωρείται ότι όλα τα διαθέσιμα φορτηγά έχουν την ίδια χωρητικότητα, η οποία είναι ίση με 200. Ακόμα θεωρείται ότι δεν υπάρχουν χρονικοί περιορισμοί.

Αποτελέσματα για τον απλό Μεμετικό Αλγόριθμο

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο έχει κόστος 875,0436 . Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (826,14) είναι:

$$\frac{875{,}0436 - 826{,}14}{826{,}14} = 5{,}92\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από οκτώ διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι οκτώ αυτές διαδρομές έχουν ως εξής:

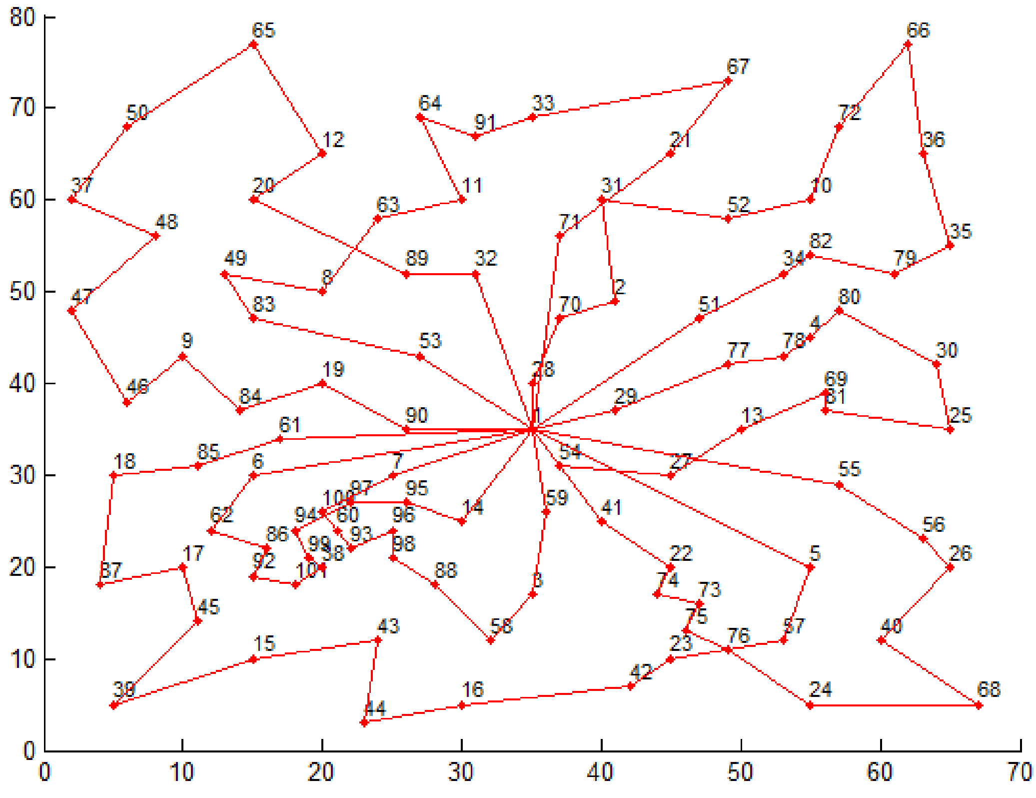
1	54	59	41	22	73	23	42	58	3	98	95	14	1					
1	88	93	38	101	86	6	94	60	96	7	1							
1	90	61	85	18	62	17	87	39	45	15	44	16	43	92	99	100	97	1
1	19	84	46	9	47	48	37	50	65	64	91	33	31	71	2	1		
1	28	70	32	89	11	63	12	20	49	83	8	53	1					
1	27	55	5	56	26	40	68	24	57	76	75	74	1					
1	13	81	69	25	30	79	35	36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

Αποτελέσματα για τον Μεμετικό Αλγόριθμο με χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο όπου έγινε χρήση της μεθόδου GRASP έχει κόστος 874.6031. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (826,14) είναι:

$$\frac{874.6031 - 826{,}14}{826{,}14} = 2{,}58\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από 8 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 8 αυτές διαδρομές έχουν ως εξής:

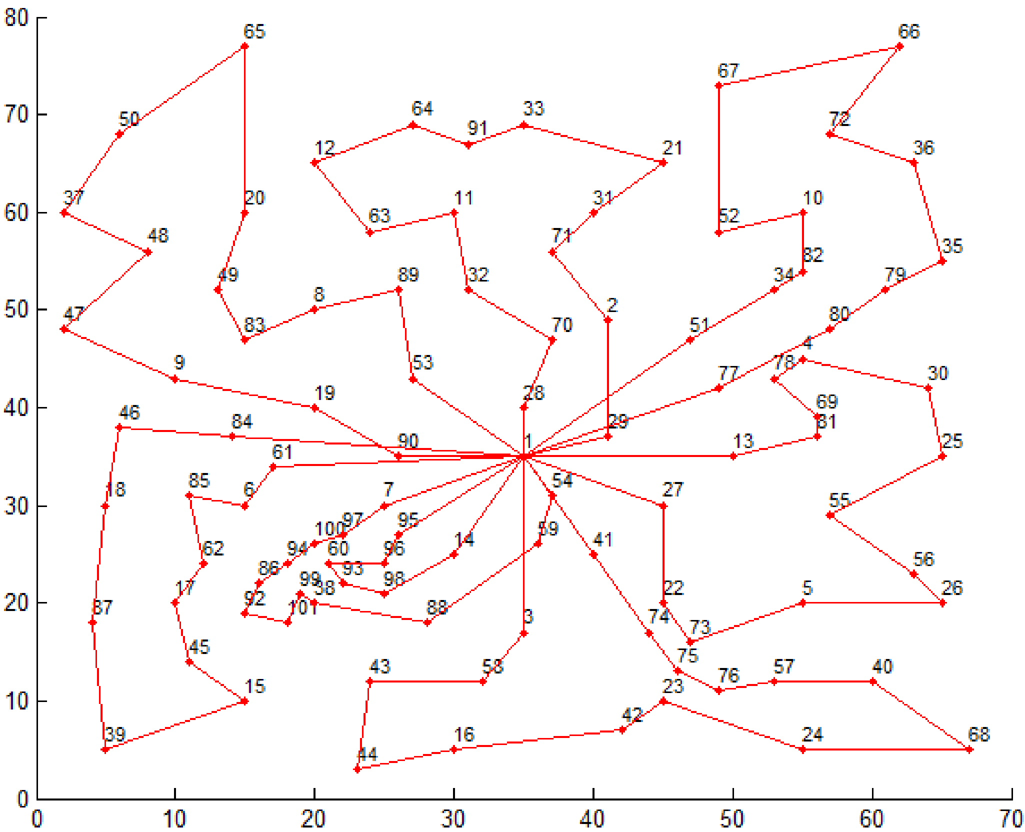
1	51	34	82	79	35	36	66	72	10	52	31	2	70	28	1	
1	29	77	78	4	80	30	25	81	69	13	27	54	1			
1	32	89	20	12	65	50	37	48	47	46	9	84	19	90	1	
1	53	83	49	8	63	11	64	91	33	67	21	71	1			
1	6	62	86	92	101	38	99	94	97	95	14	1				
1	61	85	18	87	17	45	39	15	43	44	16	42	23	57	5	1
1	55	56	26	40	68	24	76	75	73	74	22	41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

Αποτελέσματα για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών, μοιάζει πολύ με την λύση της προηγούμενης παραλλαγής του αλγορίθμου και είναι ελαφρώς βελτιωμένη. Έχει κόστος 859.8422. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (826,14) είναι:

$$\frac{859.8422 - 826{,}14}{826{,}14} = 4{,}08\%$$

Γραφικά η λύση έχει ως εξής:



Η συγκεκριμένη λύση, όπως φαίνεται και από το σχήμα, αποτελείται επίσης από οκτώ διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι οκτώ αυτές διαδρομές έχουν ως εξής:

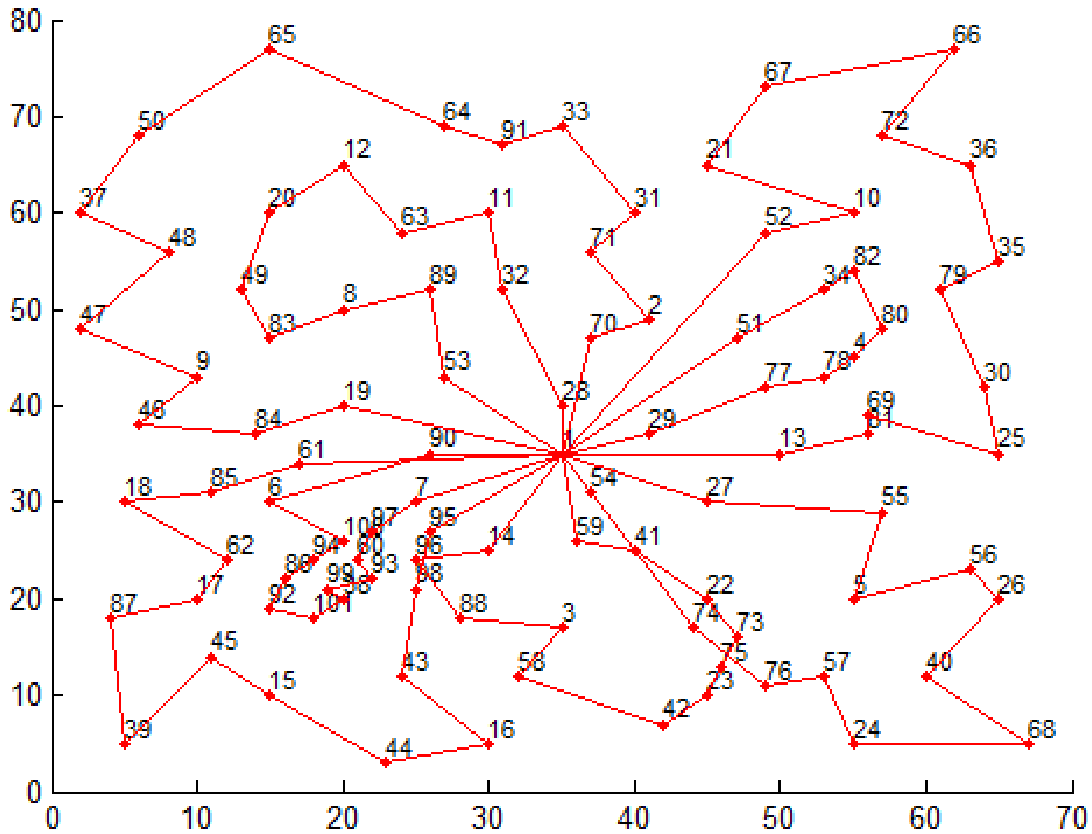
1	53	89	8	83	49	20	65	50	37	48	47	9	19	90	1	
1	84	46	18	87	39	15	45	17	62	85	6	61	1			
1	27	22	73	5	26	56	55	25	30	4	78	69	81	13	1	
1	77	80	79	35	36	72	66	67	52	10	82	34	51	1		
1	28	70	32	11	63	12	64	91	33	21	31	71	2	29	1	
1	41	74	75	76	57	40	68	24	23	42	16	44	43	58	3	1
1	54	59	88	38	99	101	92	86	94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

Αποτελέσματα για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών με χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών όπου έγινε χρήση της μεθόδου GRASP έχει κόστος 852,0737. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (826,14) είναι μόλις:

$$\frac{852{,}0737 - 826{,}14}{826{,}14} = 3{,}14\%$$

Γραφικά έχει ως εξής:



Αυτή η λύση αποτελείται από 8 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 8 αυτές διαδρομές έχουν ως εξής:

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

4.5 Παράδειγμα 4

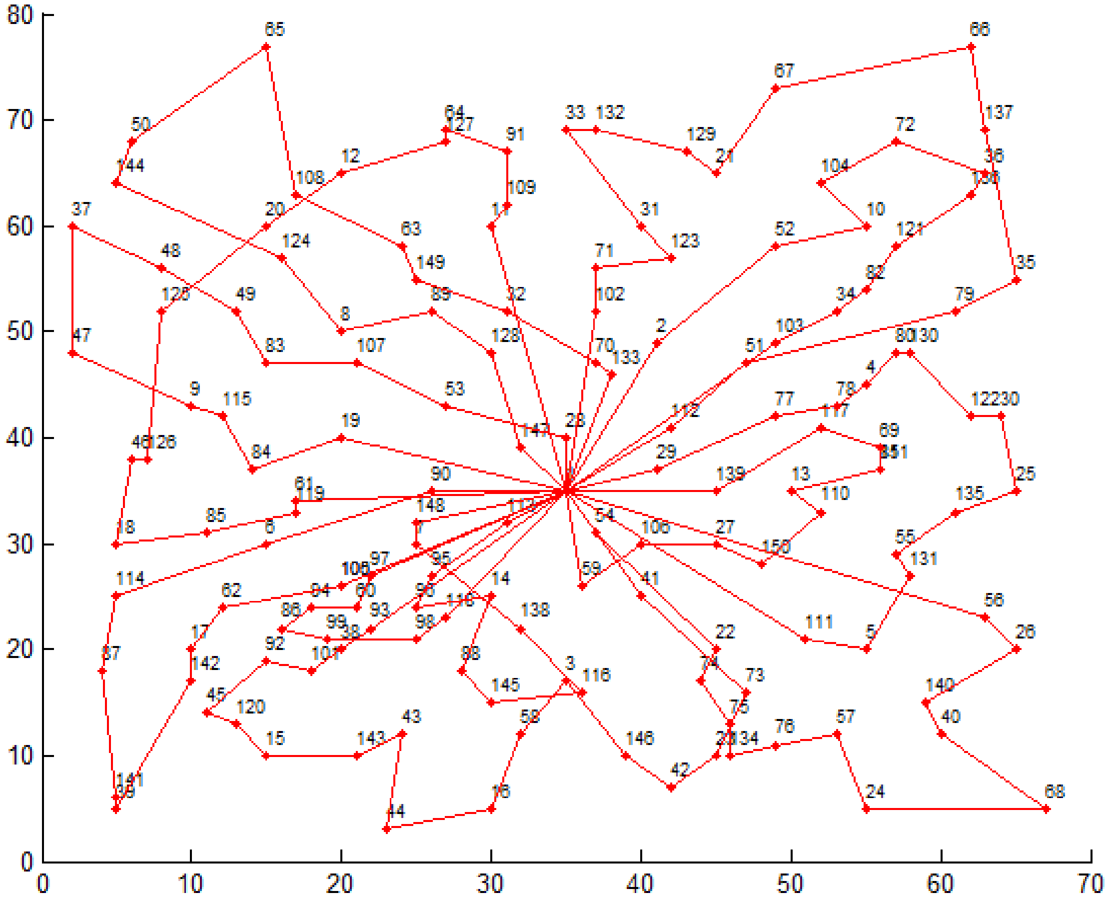
Στο 4^{o} παράδειγμα υπάρχουν 151 κόμβοι όπου ο κόμβος 1 αντιστοιχεί στην αποθήκη και οι υπόλοιποι 150 στους πελάτες. Κάθε πελάτης έχει συγκεκριμένες συντεταγμένες θέσης και συγκεκριμένη ζήτηση. Οι τιμές για τις συντεταγμένες των κόμβων και τη ζήτησή τους δίνονται στο παράρτημα. Επίσης θεωρείται ότι όλα τα διαθέσιμα φορτηγά έχουν την ίδια χωρητικότητα, η οποία είναι ίση με 200. Τέλος δεν υπάρχουν χρονικοί περιορισμοί.

<u>Αποτελέσματα για τον απλό Μεμετικό Αλγόριθμο</u>

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο έχει κόστος 1163,3428 . Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1028,42) είναι:

$$\frac{1163{,}3428 - 1028{,}42}{1028{,}42} = 13{,}12\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από δώδεκα διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι δώδεκα αυτές διαδρομές έχουν ως εξής:

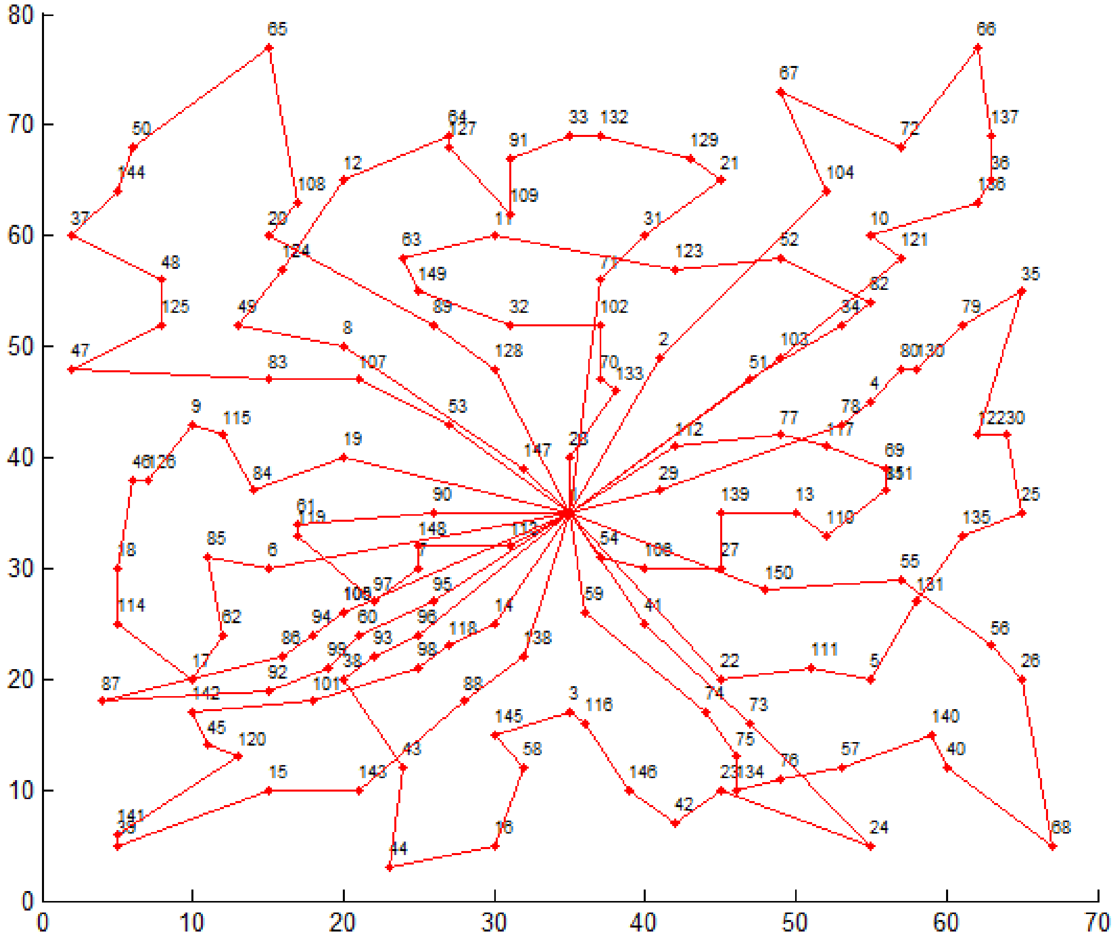
1	59	106	27	150	110	13	151	81	69	117	139	1				
1	28	53	107	83	49	48	37	47	9	115	84	19	1			
1	148	7	138	116	145	88	14	96	95	113	1					
1	54	22	74	75	134	76	57	24	68	40	140	26	56	1		
1	111	5	131	55	135	25	30	122	130	80	4	78	77	29	1	
1	2	52	10	104	72	36	136	121	82	34	103	1				
1	112	51	79	35	137	66	67	21	129	132	33	31	123	71	102	1
1	133	70	32	149	63	108	65	50	144	124	8	89	128	147	1	
1	41	73	23	42	146	3	58	16	44	43	143	15	120	45	…	
…	92	101	38	93	1											
1	11	109	91	64	127	12	20	125	126	46	18	85	119	61	1	
1	90	6	114	87	141	39	142	17	62	100	105	1				
1	97	60	94	86	99	98	118	1								

Αποτελέσματα για τον Μεμετικό Αλγόριθμο με χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο όπου έγινε χρήση της μεθόδου GRASP έχει κόστος 1160,9520. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1028,42) είναι:

$$\frac{1160{,}9520 - 1028{,}42}{1028{,}42} = 12{,}89\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από 12 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 12 αυτές διαδρομές έχουν ως εξής:

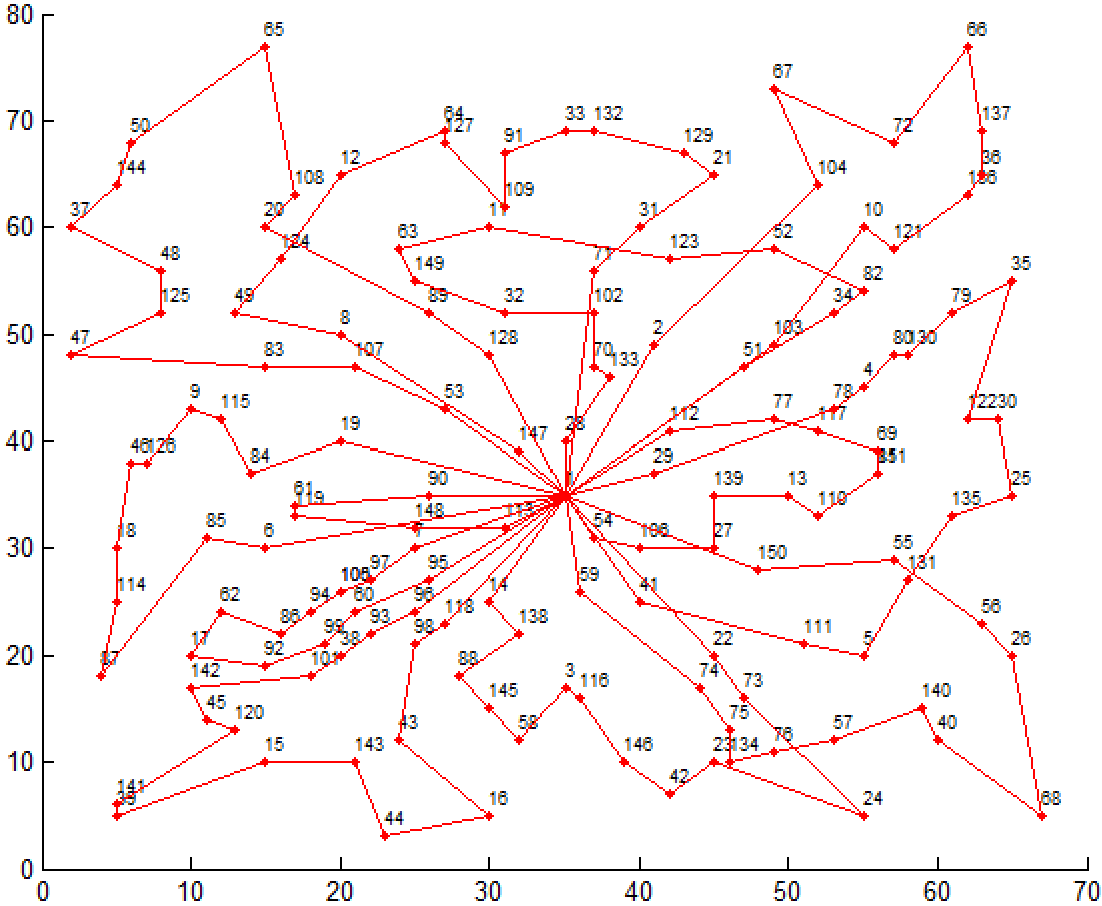
1	147	8	49	124	12	64	127	109	91	33	132	129	21	31	71	1	
1	2	104	67	72	66	137	36	136	10	121	103	1					
1	112	77	117	69	81	151	110	13	139	27	106	54	1				
1	150	55	56	26	68	40	140	57	76	134	75	74	59	1			
1	22	111	5	131	135	25	30	122	35	79	130	80	4	78	29	1	
1	51	34	82	52	123	11	63	149	32	102	70	133	28	1			
1	128	89	20	108	65	50	144	37	48	125	47	83	107	53	1		
1	19	84	115	9	126	46	18	114	17	62	85	6	1				
1	95	60	99	92	87	86	94	100	105	1							
1	96	93	38	43	44	16	58	145	3	116	146	42	23	24	73	41	1
1	14	118	98	101	142	45	120	141	39	15	143	88	138	1			
1	113	148	7	97	119	61	90	1									

Αποτελέσματα για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών, μοιάζει πολύ με την λύση της προηγούμενης παραλλαγής του αλγορίθμου και είναι ελαφρώς βελτιωμένη. Έχει κόστος 1137,3237. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1028,42) είναι:

$$\frac{1137{,}3237 - 1028{,}42}{1028{,}42} = 10{,}59\%$$

Γραφικά η λύση έχει ως εξής:



Η συγκεκριμένη λύση, όπως φαίνεται και από το σχήμα, αποτελείται επίσης από 12 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 12 αυτές διαδρομές έχουν ως εξής:

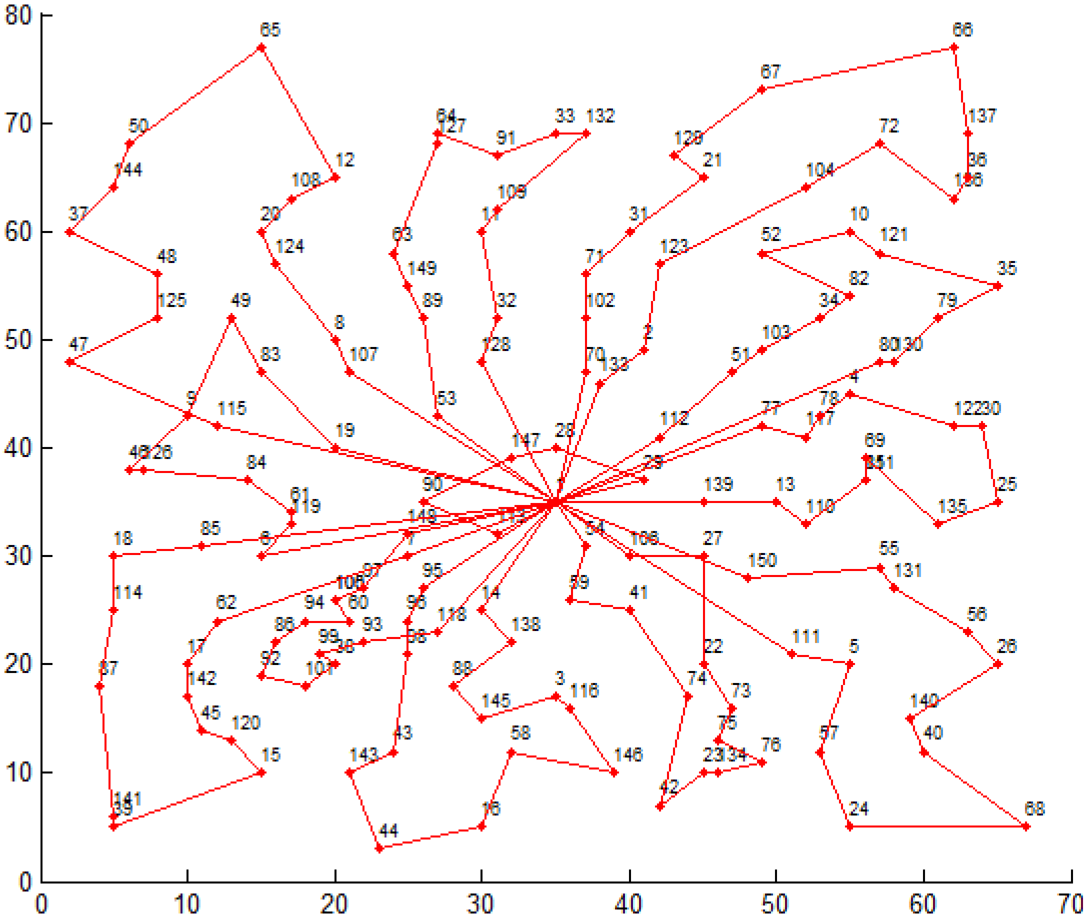
1	147	8	49	124	12	64	127	109	91	33	132	129	21	31	71	1	
1	2	104	67	72	66	137	36	136	121	10	103	1					
1	29	78	4	80	130	79	35	122	30	25	135	131	5	111	41	1	
1	59	74	75	134	76	57	140	40	68	26	56	55	150	1			
1	54	106	27	139	13	110	151	81	69	117	77	112	1				
1	51	34	82	52	123	11	63	149	32	102	70	133	28	1			
1	128	89	20	108	65	50	144	37	48	125	47	83	107	53	1		
1	19	84	115	9	126	46	18	114	87	85	6	1					
1	95	60	99	92	17	62	86	94	100	105	97	7	1				
1	96	93	38	101	142	45	120	141	39	15	143	44	16	43	98	118	1
1	14	138	88	145	58	3	116	146	42	23	24	73	22	1			
1	113	148	119	61	90	1											

<u>Αποτελέσματα για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών με χρήση της μεθόδου GRASP</u>

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών όπου έγινε χρήση της μεθόδου GRASP έχει κόστος 1072,8858. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (835,26) είναι μόλις:

$$\frac{1072{,}8858 - 1028{,}42}{1028{,}42} = 4{,}32\%$$

Γραφικά έχει ως εξής:



Αυτή η λύση αποτελείται από δώδεκα διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι δώδεκα αυτές διαδρομές έχουν ως εξής:

1	133	2	123	104	72	136	36	137	66	67	129	21	31	71	102	70	1
1	80	130	79	35	121	10	52	82	34	103	51	112	1				
1	139	13	110	81	151	69	135	25	30	122	4	78	117	77	1		
1	111	5	57	24	68	40	140	26	56	131	55	150	1				
1	106	27	22	73	75	76	134	23	42	74	41	59	54	1			
1	95	96	98	43	143	44	16	58	146	116	3	145	88	138	14	1	
1	7	62	17	142	45	120	15	39	141	87	114	18	85	1			
1	115	47	125	48	37	144	50	65	12	108	20	124	8	107	1		
1	53	89	149	63	127	64	91	33	132	109	11	32	128	1			
1	19	83	49	9	46	126	84	61	119	6	1						
1	148	97	105	100	60	94	86	92	101	38	99	93	118	1			
1	113	90	147	28	29	1											

4.6 Παράδειγμα 5

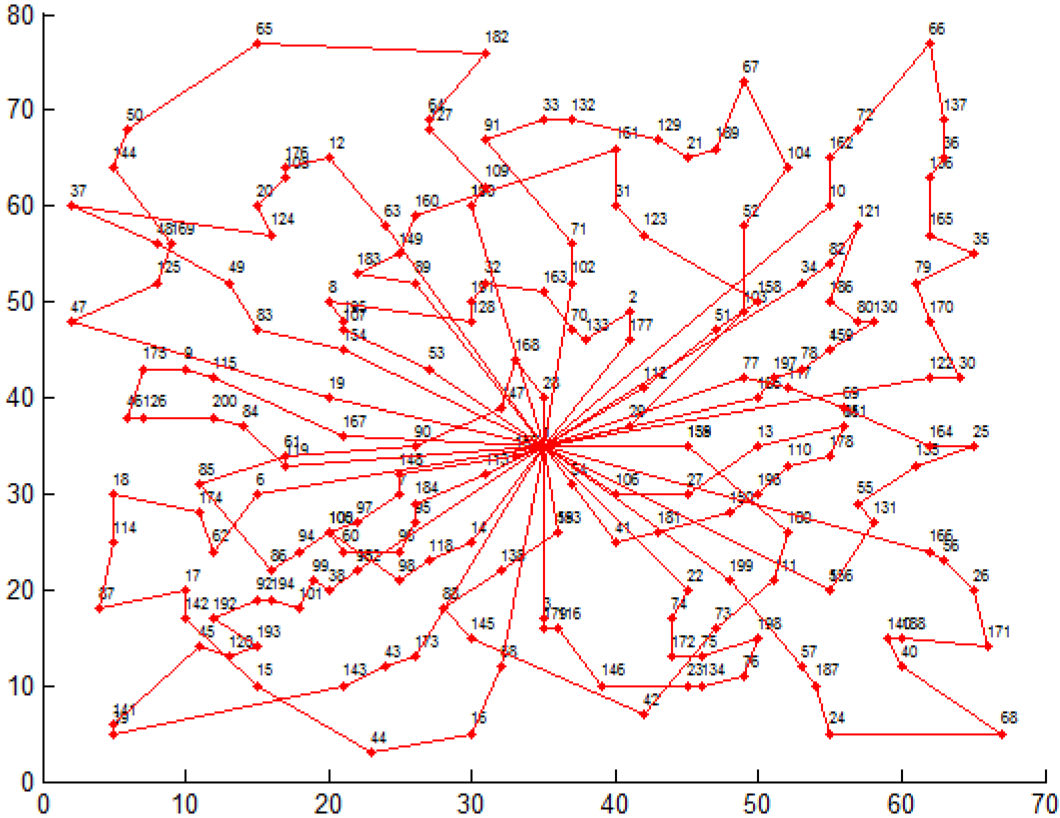
Σε αυτό το παράδειγμα υπάρχουν 200 κόμβοι. Ο κόμβος 1 αντιστοιχεί στην αποθήκη και οι υπόλοιποι 199 στους πελάτες. Κάθε πελάτης έχει συγκεκριμένες συντεταγμένες θέσης και συγκεκριμένη ζήτηση, ακριβή νούμερα για τα οποία δίνονται στο παράρτημα. Θεωρείται ότι όλα τα διαθέσιμα φορτηγά έχουν την ίδια χωρητικότητα, η οποία είναι ίση με 200. Ακόμα θεωρείται ότι δεν υπάρχουν χρονικοί περιορισμοί.

<u>Αποτελέσματα για τον απλό Μεμετικό Αλγόριθμο</u>

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο έχει κόστος 1485,2177. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1291,29) είναι:

$$\frac{1485{,}2177 - 1291{,}29}{1291{,}29} = 15{,}02\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από 17 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 17 αυτές διαδρομές έχουν ως εξής:

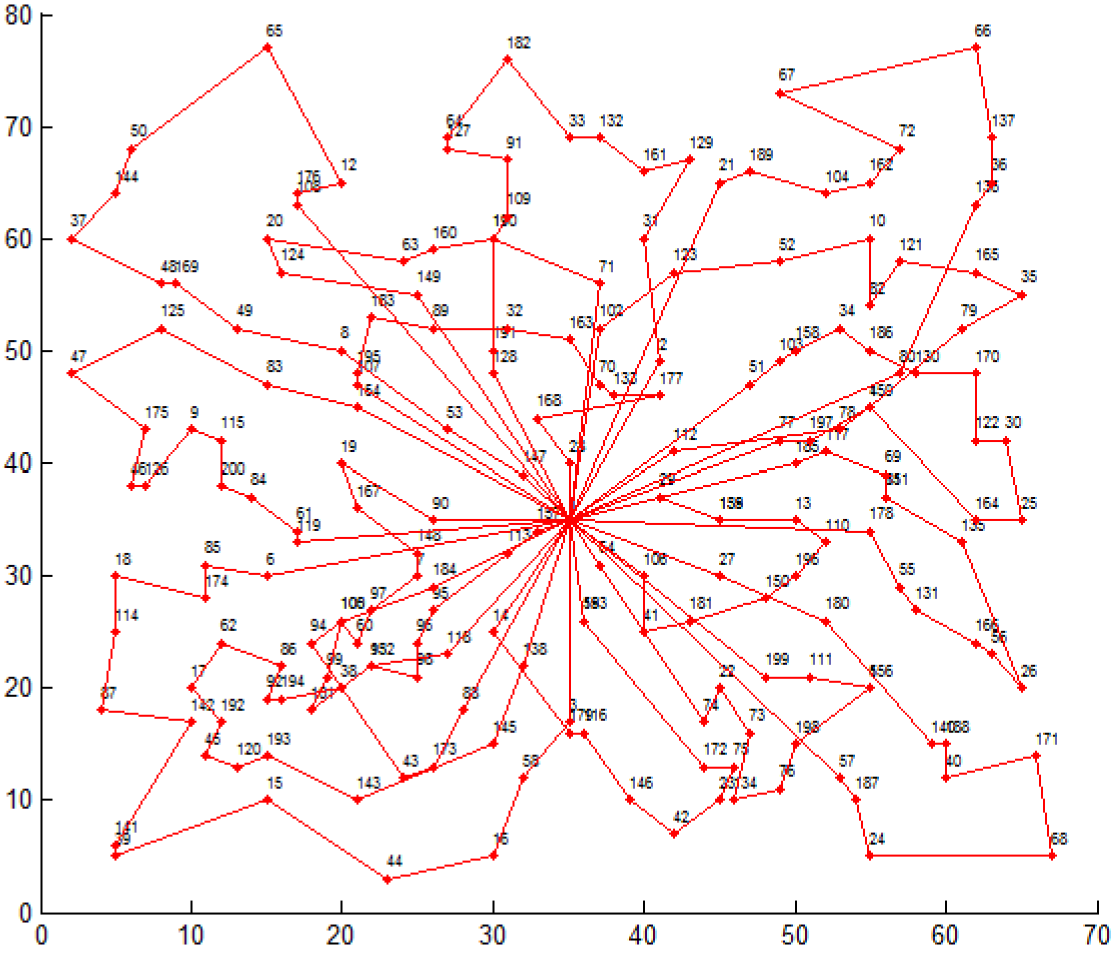
1	54	41	181	150	196	110	178	151	81	13	27	106	1				
1	185	197	78	4	159	130	80	186	121	82	34	1					
1	122	30	170	79	35	165	136	36	137	66	72	162	10	1			
1	177	2	133	70	163	32	191	128	8	195	107	53	1				
1	63	12	176	108	20	124	37	48	49	83	154	1					
1	155	139	180	111	73	42	145	88	138	153	59	1					
1	29	103	52	104	67	189	21	129	132	33	91	71	102	1			
1	152	93	38	99	101	194	92	192	193	120	45	141	39	143	43	173	1
1	14	118	98	100	94	86	85	61	90	147	168	28	1				
1	113	184	95	96	60	105	97	7	148	157	1						
1	112	51	158	123	31	161	160	149	183	89	1						
1	190	11	109	127	64	182	65	50	144	169	125	47	19	1			
1	119	84	200	126	46	175	9	115	167	1							
1	6	62	174	18	114	87	17	142	15	44	16	58	1				
1	22	74	172	75	198	76	134	23	146	116	179	3	1				
1	199	57	187	24	68	40	140	188	171	26	56	166	1				
1	5	156	131	55	135	25	164	69	117	77	1						

Αποτελέσματα για τον Μεμετικό Αλγόριθμο με χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο όπου έγινε χρήση της μεθόδου GRASP έχει κόστος 1471,0816. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1291,29) είναι:

$$\frac{1471{,}0816 - 1291{,}29}{1291{,}29} = 13{,}92\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από δεκαεπτά διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι δεκαεπτά αυτές διαδρομές έχουν ως εξής:

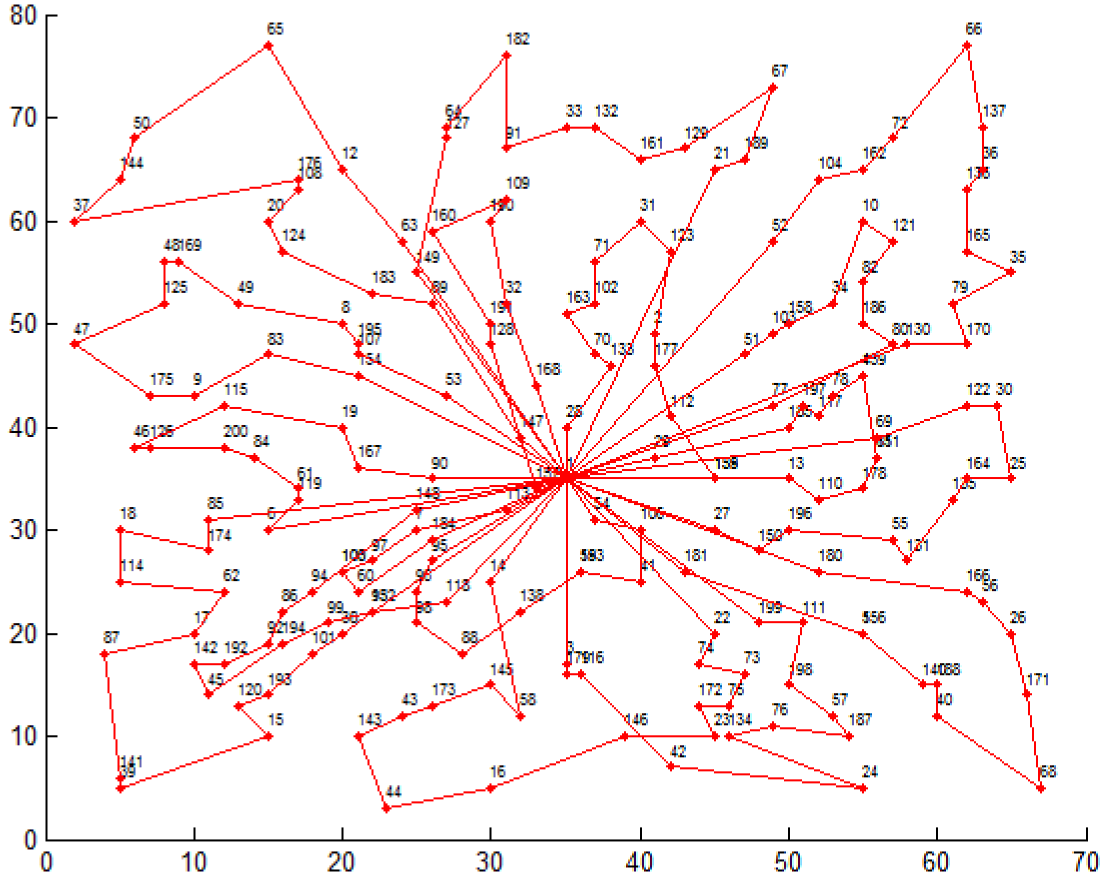
1	157	113	95	96	98	152	101	99	100	184	1					
1	27	180	140	188	40	171	68	24	187	57	1					
1	199	111	5	156	198	76	134	73	22	74	54	1				
1	59	153	172	75	23	42	146	116	179	14	1					
1	88	173	43	94	105	60	97	7	148	167	19	90	1			
1	107	195	183	89	32	163	70	133	177	168	28	1				
1	185	117	69	151	81	135	26	56	166	131	55	178	1			
1	112	78	4	79	35	165	121	82	10	52	123	102	1			
1	108	176	12	65	50	144	37	48	169	49	8	53	147	1		
1	154	83	125	47	175	46	126	9	115	200	84	61	119	1		
1	6	85	174	18	114	87	142	141	39	15	44	16	58	3	1	
1	118	93	38	194	92	86	62	17	192	45	120	193	143	145	138	1
1	106	41	181	150	196	110	13	155	139	29	1					
1	80	136	36	137	66	67	72	162	104	189	21	1				
1	71	11	109	91	127	64	182	33	132	161	129	31	2	1		
1	77	197	159	164	25	30	122	170	130	186	34	158	103	51	1	
1	149	124	20	63	160	190	191	128	1							

Αποτελέσματα για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών, μοιάζει πολύ με την λύση της προηγούμενης παραλλαγής του αλγορίθμου και είναι ελαφρώς βελτιωμένη. Έχει κόστος 1412,3718. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1291,29) είναι:

$$\frac{1412{,}3718 - 1291{,}29}{1291{,}29} = 9{,}38\%$$

Γραφικά η λύση έχει ως εξής:



Η συγκεκριμένη λύση, όπως φαίνεται και από το σχήμα, αποτελείται επίσης από 17 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 17 αυτές διαδρομές έχουν ως εξής:

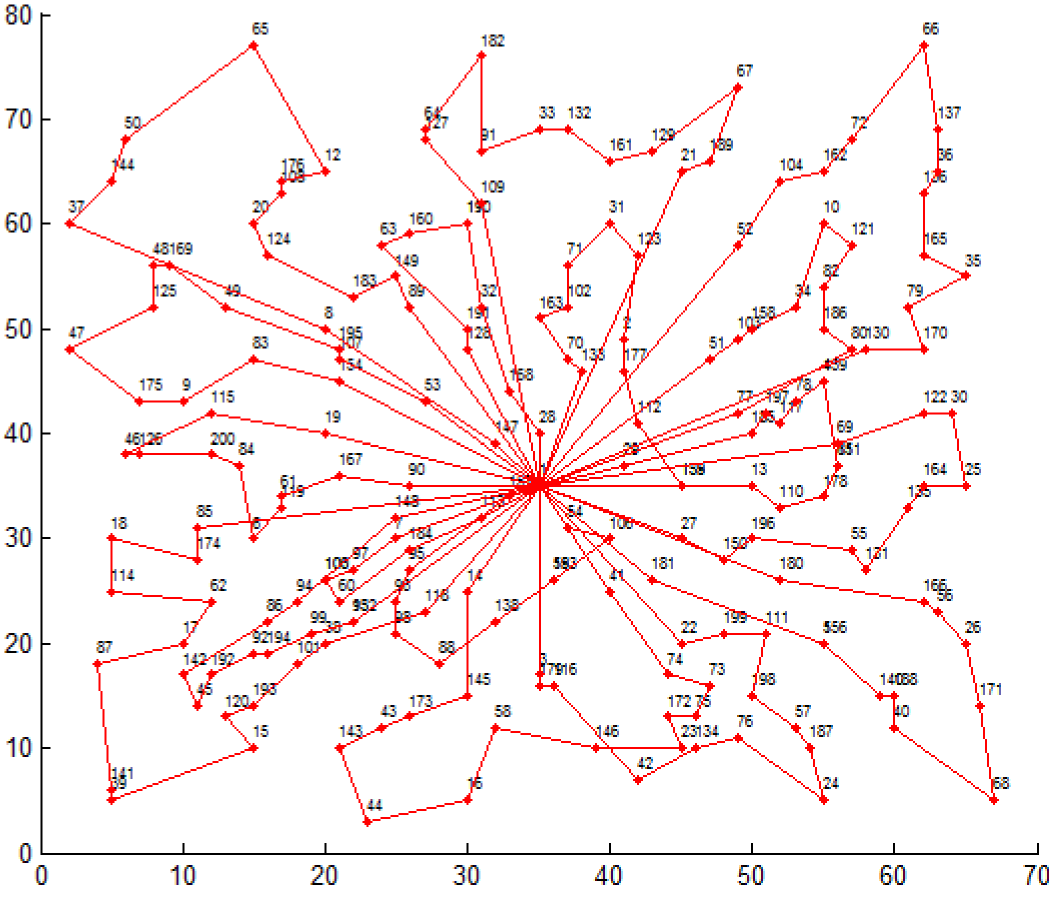
1	147	8	37	144	50	65	12	176	108	20	124	183	149	89	1	
1	109	127	64	182	91	33	132	161	129	67	189	21	1			
1	52	104	162	72	66	137	36	136	165	35	79	170	130	1		
1	51	103	158	34	10	121	82	186	80	77	1					
1	13	110	178	81	151	159	4	78	117	197	185	29	1			
1	155	139	112	177	2	123	31	71	102	163	70	133	1			
1	69	122	30	25	164	135	131	55	196	150	27	1				
1	181	156	5	140	188	40	68	171	26	56	166	180	1			
1	41	74	73	75	172	23	146	58	16	44	143	43	173	145	14	1
1	3	179	116	42	134	76	24	187	57	198	111	199	22	1		
1	154	83	9	175	47	125	48	169	49	195	107	53	1			
1	128	191	63	160	190	11	32	168	28	1						
1	157	95	96	98	88	138	59	153	106	54	1					
1	152	93	99	194	92	192	45	142	86	94	148	1				
1	90	167	61	119	6	84	200	126	46	115	19	1				
1	85	174	18	114	62	17	87	141	39	15	120	193	101	38	118	1
1	113	184	60	100	105	97	7	1								

Αποτελέσματα για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών με χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών όπου έγινε χρήση της μεθόδου GRASP έχει κόστος 1387,9302. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1291,29) είναι μόλις:

$$\frac{1387{,}9302 - 1291{,}29}{1291{,}29} = 7{,}48\%$$

Γραφικά έχει ως εξής:



Αυτή η λύση αποτελείται από δεκαεπτά διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι δεκαεπτά αυτές διαδρομές έχουν ως εξής:

1	147	8	37	144	50	65	12	176	108	20	124	183	149	89	1
1	109	127	64	182	91	33	132	161	129	67	189	21	1		
1	52	104	162	72	66	137	36	136	165	35	79	170	130	1	
1	51	103	158	34	10	121	82	186	80	77	1				
1	13	110	178	81	151	159	4	78	117	197	185	29	1		
1	155	139	112	177	2	123	31	71	102	163	70	133	1		
1	69	122	30	25	164	135	131	55	196	150	27	1			

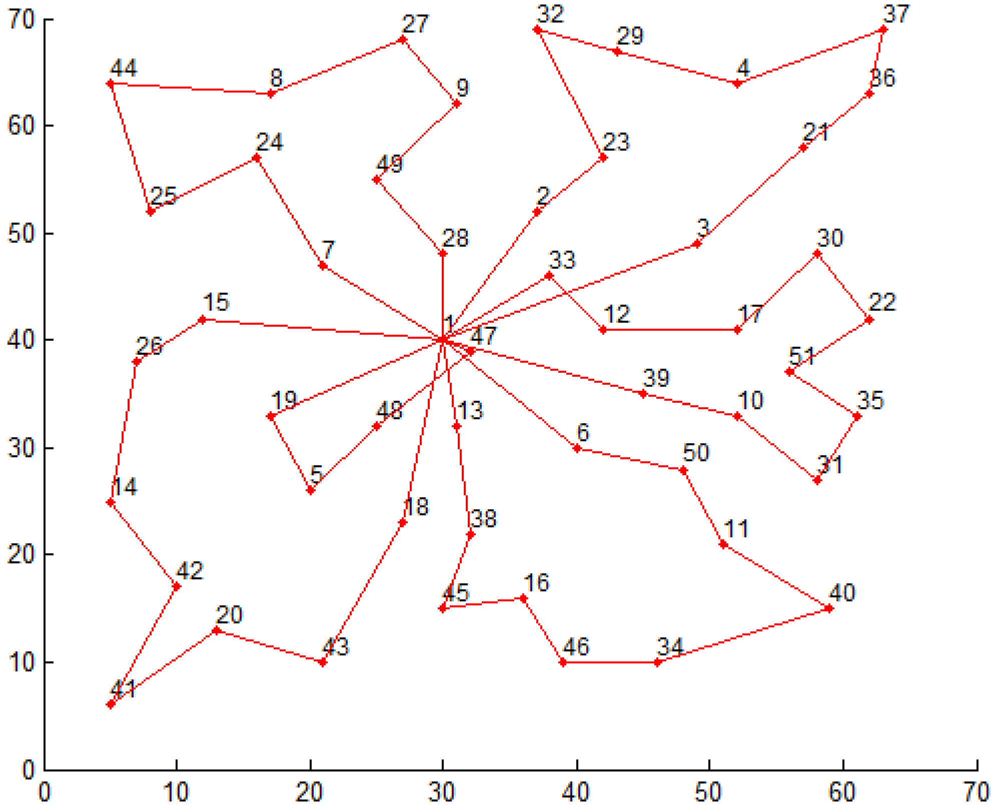
1	181	156	5	140	188	40	68	171	26	56	166	180	1			
1	41	74	73	75	172	23	146	58	16	44	143	43	173	145	14	1
1	3	179	116	42	134	76	24	187	57	198	111	199	22	1		
1	154	83	9	175	47	125	48	169	49	195	107	53	1			
1	128	191	63	160	190	11	32	168	28	1						
1	157	95	96	98	88	138	59	153	106	54	1					
1	152	93	99	194	92	192	45	142	86	94	148	1				
1	90	167	61	119	6	84	200	126	46	115	19	1				
1	85	174	18	114	62	17	87	141	39	15	120	193	101	38	118	1
1	113	184	60	100	105	97	7	1								

4.7 Παράδειγμα 6

Σε αυτό το παράδειγμα υπάρχουν 51 κόμβοι. Ο κόμβος 1 αντιστοιχεί στην αποθήκη και οι υπόλοιποι 50 στους πελάτες. Κάθε πελάτης έχει συγκεκριμένες συντεταγμένες θέσης και συγκεκριμένη ζήτηση, ακριβή νούμερα για τα οποία δίνονται στο παράρτημα. Θεωρείται ότι όλα τα διαθέσιμα φορτηγά έχουν την ίδια χωρητικότητα, η οποία είναι ίση με 160. Τέλος, καμία διαδρομή δεν πρέπει να έχει συνολικό μήκος πάνω από 200 μονάδες.

Για το συγκεκριμένο παράδειγμα, τα αποτελέσματα και των τεσσάρων αλγορίθμων που υλοποιήθηκαν στα πλαίσια αυτής της διπλωματικής εργασίας ήταν ίδια μεταξύ τους. Ίδια ήταν επίσης η λύση μας με το μέχρις στιγμής βέλτιστο αποτέλεσμα που έχει βρεθεί από ανάλογες έρευνες σε παγκόσμια κλίμακα για το $6^{ο}$ αυτό παράδειγμα. Η βέλτιστη λύση λοιπόν που βρήκαμε έχει κόστος ίσο με 555,4302 και γραφικά έχει ως εξής:

Γραφικά έχει ως εξής:



Η συγκεκριμένη λύση, όπως φαίνεται και από το σχήμα, αποτελείται από έξι διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι έξι αυτές διαδρομές έχουν ως εξής:

1	33	12	17	30	22	51	35	31	10	39	1
1	6	50	11	40	34	46	16	45	38	13	1
1	18	43	20	41	42	14	26	15	1		
1	7	24	25	44	8	27	9	49	28	1	
1	2	23	32	29	4	37	36	21	3	1	
1	19	5	48	47	1						

4.8 Παράδειγμα 7

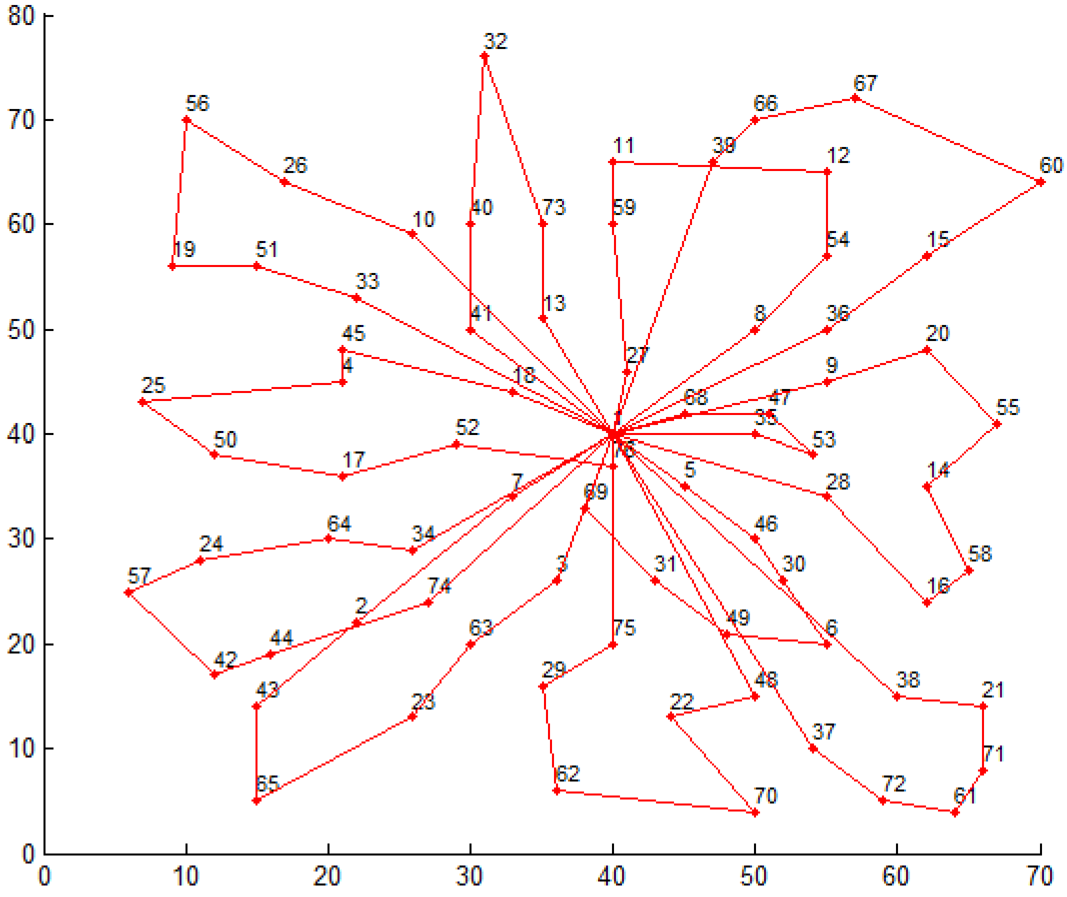
Σε αυτό το παράδειγμα υπάρχουν 76 κόμβοι. Ο κόμβος 1 αντιστοιχεί στην αποθήκη και οι υπόλοιποι 75 στους πελάτες. Κάθε πελάτης έχει συγκεκριμένες συντεταγμένες θέσης και συγκεκριμένη ζήτηση, ακριβή νούμερα για τα οποία δίνονται στο παράρτημα. Θεωρείται ότι όλα τα διαθέσιμα φορτηγά έχουν την ίδια χωρητικότητα, η οποία είναι ίση με 140. Επίσης, καμία διαδρομή δεν πρέπει να έχει συνολικό μήκος πάνω από 160.

<u>Αποτελέσματα για τον απλό Μεμετικό Αλγόριθμο</u>

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο έχει κόστος 944.1599 . Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (909,68) είναι:

$$\frac{944{,}1599 - 909{,}68}{909{,}68} = 3{,}79\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από 12 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 12 αυτές διαδρομές έχουν ως εξής:

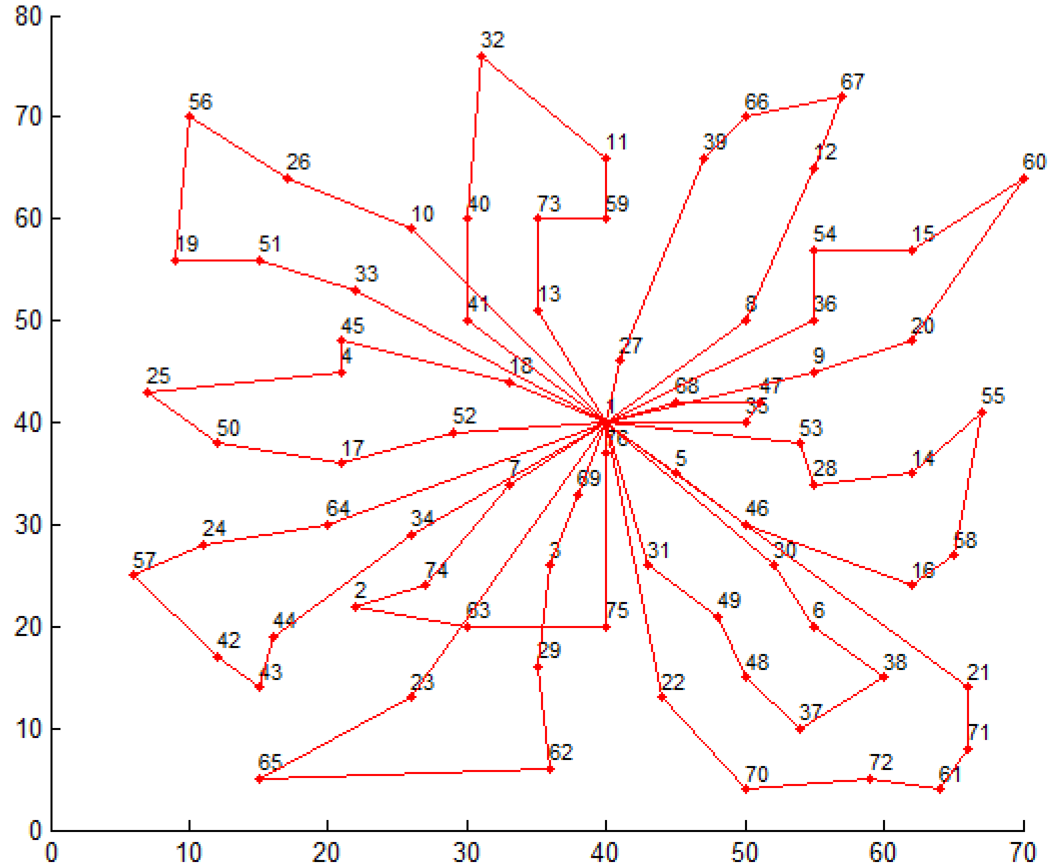
1	69	31	49	6	30	46	5	1	
1	7	2	43	65	23	63	3	1	
1	75	29	62	70	22	48	1		
1	38	21	71	61	72	37	1		
1	10	26	56	19	51	33	1		
1	41	40	32	73	13	1			
1	28	16	58	14	55	20	9	1	
1	74	44	42	57	24	64	34	1	
1	76	52	17	50	25	4	45	18	1
1	27	59	11	12	54	8	1		
1	39	66	67	60	15	36	1		
1	35	53	47	68	1				

Αποτελέσματα για τον Μεμετικό Αλγόριθμο με χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο όπου έγινε χρήση της μεθόδου GRASP έχει κόστος 942,1404 . Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (909,68) είναι:

$$\frac{942{,}1404 - 909{,}68}{909{,}68} = 3{,}57\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από δώδεκα διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι δώδεκα αυτές διαδρομές έχουν ως εξής:

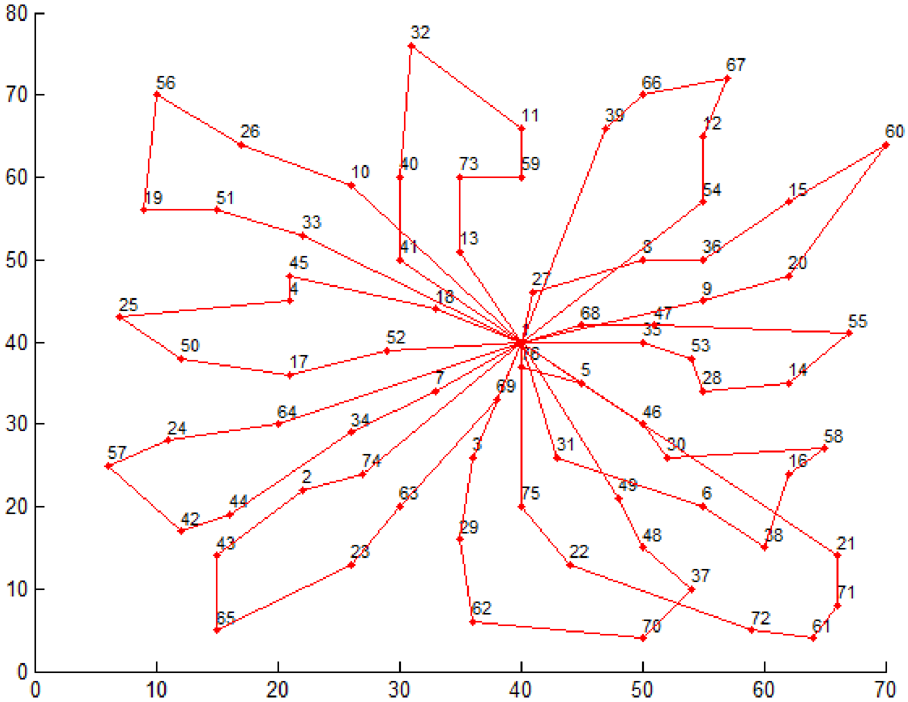
1	27	39	66	67	12	8	1		
1	41	40	32	11	59	73	13	1	
1	69	3	29	62	65	23	1		
1	64	24	57	42	43	44	34	1	
1	76	75	63	2	74	7	1		
1	30	6	38	37	48	49	31	1	
1	5	46	16	58	55	14	28	53	1
1	36	54	15	60	20	9	1		
1	33	51	19	56	26	10	1		
1	22	70	72	61	71	21	1		
1	18	45	4	25	50	17	52	1	
1	68	47	35	1					

Αποτελέσματα για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών, μοιάζει πολύ με την λύση της προηγούμενης παραλλαγής του αλγορίθμου και είναι ελαφρώς βελτιωμένη. Έχει κόστος 929.0642. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (909,68) είναι:

$$\frac{929.0642 - 909,68}{909,68} = 2,13\%$$

Γραφικά η λύση έχει ως εξής:



Η συγκεκριμένη λύση, όπως φαίνεται και από το σχήμα, αποτελείται επίσης από 12 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 12 αυτές διαδρομές έχουν ως εξής:

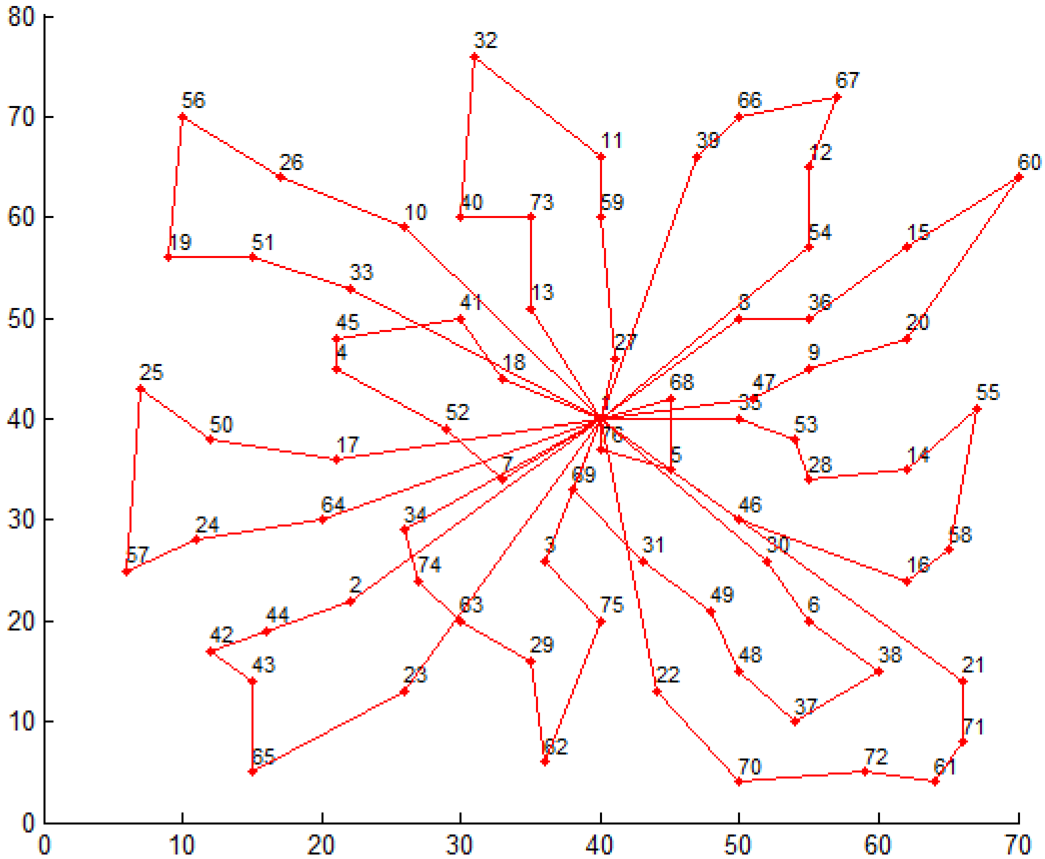
1	69	63	23	65	43	2	74	1
1	3	29	62	70	37	48	49	1
1	75	22	72	61	71	21	1	
1	39	66	67	12	54	1		
1	35	53	28	14	55	47	68	1
1	9	20	60	15	36	8	27	1
1	46	30	58	16	38	6	31	1
1	7	34	44	42	57	24	64	1
1	52	17	50	25	4	45	18	1
1	13	73	59	11	32	40	41	1
1	33	51	19	56	26	10	1	
1	5	76	1					

Αποτελέσματα για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών με χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών όπου έγινε χρήση της μεθόδου GRASP έχει κόστος 926.9197. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (909,68) είναι μόλις:

$$\frac{926{,}9197 - 909{,}68}{909{,}68} = 1{,}90\%$$

Γραφικά έχει ως εξής:



Αυτή η λύση αποτελείται από δώδεκα διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι δώδεκα αυτές διαδρομές έχουν ως εξής:

1	13	73	40	32	11	59	27	1	
1	47	9	20	60	15	36	8	1	
1	54	12	67	66	39	1			
1	7	52	4	45	41	18	1		
1	10	26	56	19	51	33	1		
1	17	50	25	57	24	64	1		
1	2	44	42	43	65	23	1		
1	3	75	62	29	63	74	34	1	
1	69	31	49	48	37	38	6	30	1
1	22	70	72	61	71	21	1		
1	46	16	58	55	14	28	53	35	1
1	76	5	68	1					

4.9 Παράδειγμα 8

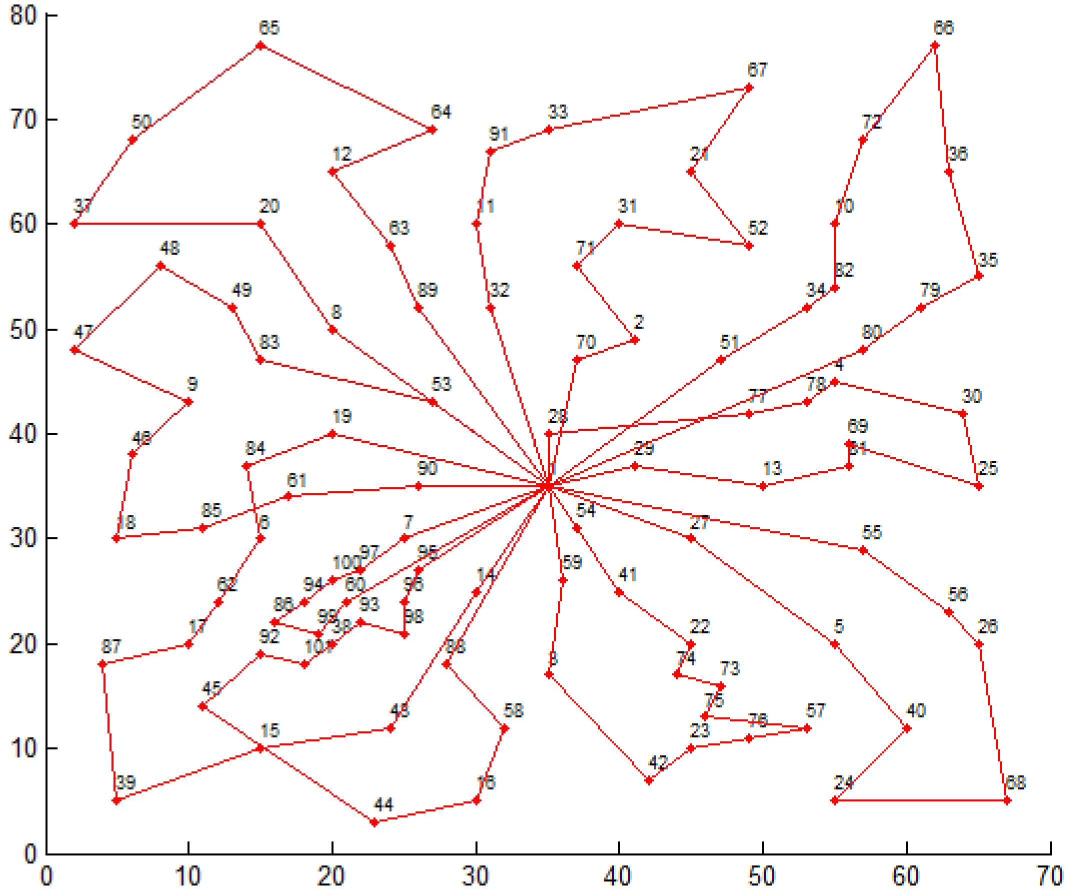
Στο 8^{ο} παράδειγμα υπάρχουν 101 κόμβοι όπου ο κόμβος 1 αντιστοιχεί στην αποθήκη και οι υπόλοιποι 100 στους πελάτες. Κάθε πελάτης έχει συγκεκριμένες συντεταγμένες θέσης και συγκεκριμένη ζήτηση. Οι τιμές για τις συντεταγμένες των κόμβων και τη ζήτησή τους δίνονται στο παράρτημα. Επίσης θεωρείται ότι όλα τα διαθέσιμα φορτηγά έχουν την ίδια χωρητικότητα, η οποία είναι ίση με 200. Τέλος, καμία διαδρομή δεν πρέπει να έχει συνολικό μήκος πάνω από 230 μονάδες.

<u>Αποτελέσματα για τον απλό Μεμετικό Αλγόριθμο</u>

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο έχει κόστος 948,1406. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (865,94) είναι:

$$\frac{948{,}1406 - 865{,}94}{865{,}94} = 9{,}49\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από 10 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 10 αυτές διαδρομές έχουν ως εξής:

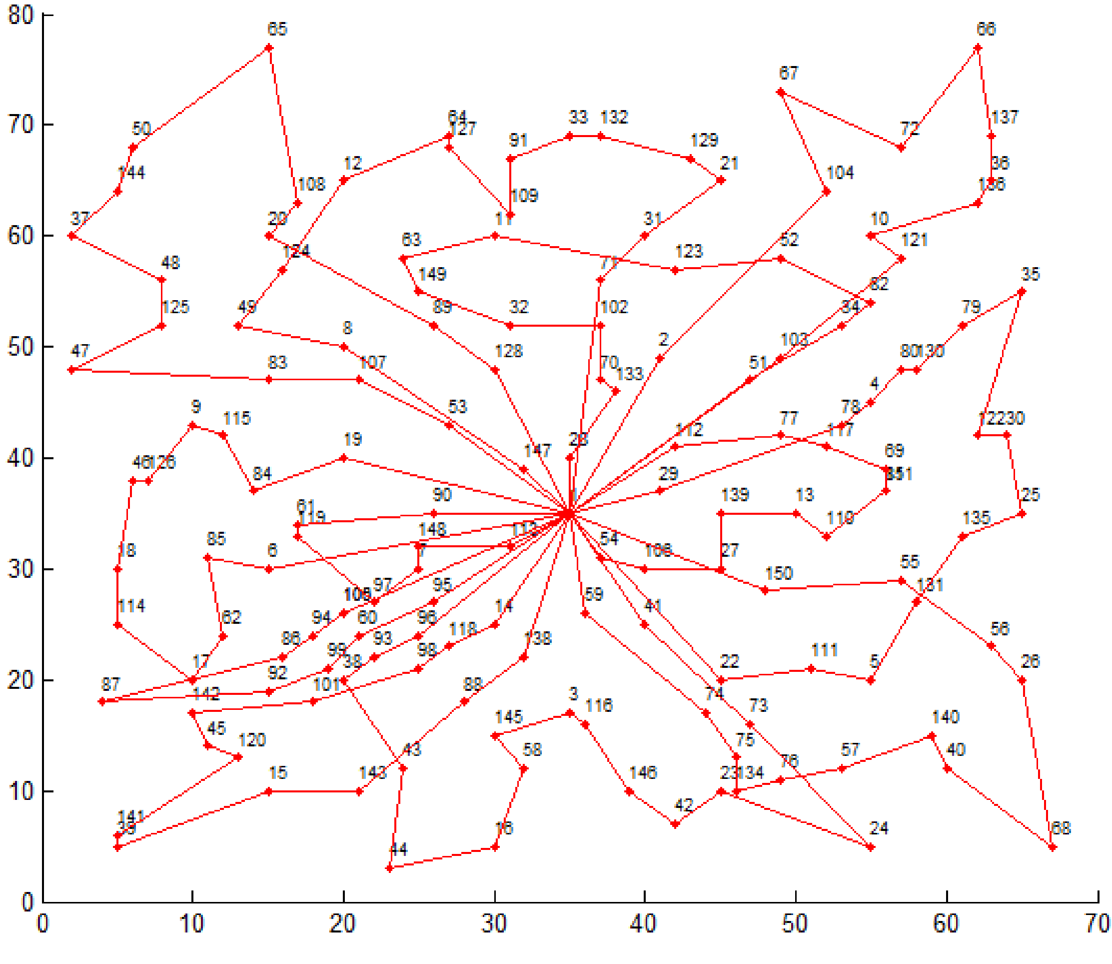
1	29	13	81	69	25	30	4	78	77	28	1		
1	80	79	35	36	66	72	10	82	34	51	1		
1	55	56	26	68	24	40	5	27	1				
1	89	63	12	64	65	50	37	20	8	1			
1	88	58	16	44	45	92	101	38	93	98	96	95	1
1	59	3	42	23	76	57	75	73	74	22	41	54	1
1	32	11	91	33	67	21	52	31	71	2	70	1	
1	53	83	49	48	47	9	46	18	85	61	90	1	
1	19	84	6	62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

Αποτελέσματα για τον Μεμετικό Αλγόριθμο με χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο όπου έγινε χρήση της μεθόδου GRASP έχει κόστος 1160,9520. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1028,42) είναι:

$$\frac{1160{,}9520 - 1028{,}42}{1028{,}42} = 12{,}89\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από 12 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 12 αυτές διαδρομές έχουν ως εξής:

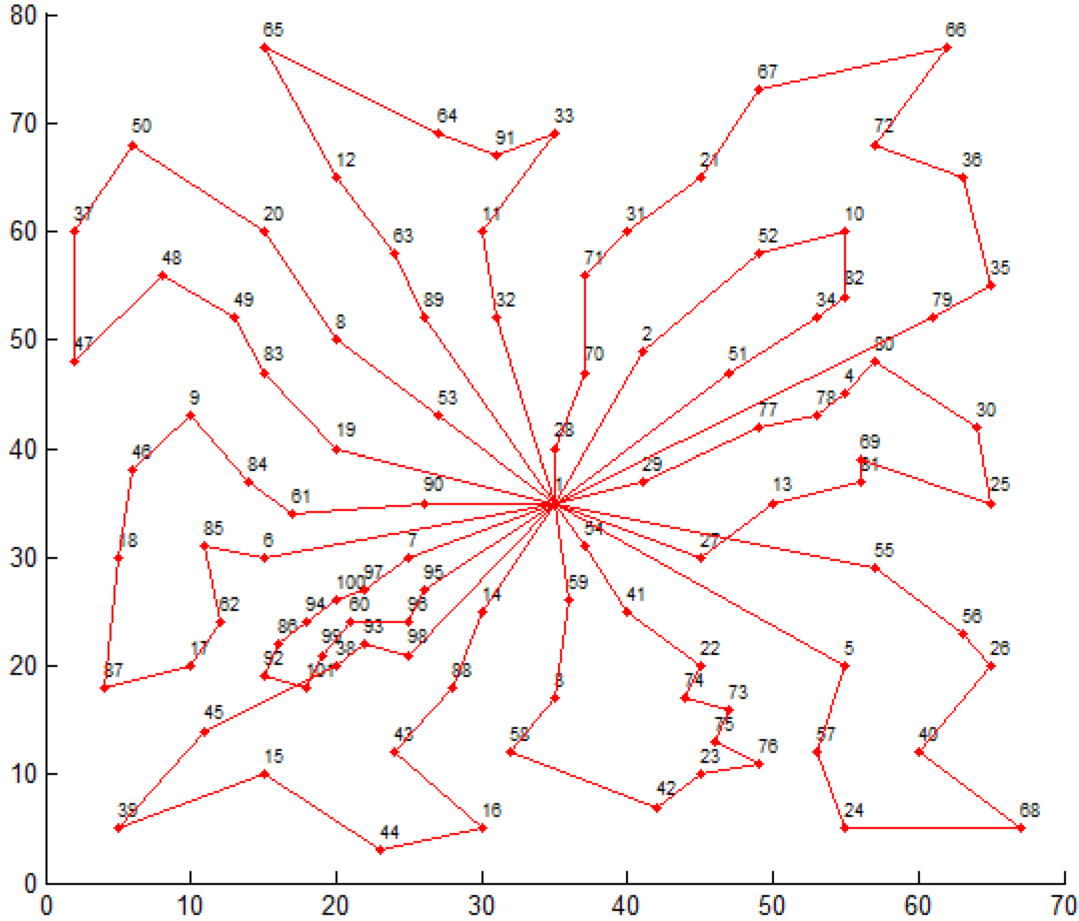
1	147	8	49	124	12	64	127	109	91	33	132	129	21	31	71	1	
1	2	104	67	72	66	137	36	136	10	121	103	1					
1	112	77	117	69	81	151	110	13	139	27	106	54	1				
1	150	55	56	26	68	40	140	57	76	134	75	74	59	1			
1	22	111	5	131	135	25	30	122	35	79	130	80	4	78	29	1	
1	51	34	82	52	123	11	63	149	32	102	70	133	28	1			
1	128	89	20	108	65	50	144	37	48	125	47	83	107	53	1		
1	19	84	115	9	126	46	18	114	17	62	85	6	1				
1	95	60	99	92	87	86	94	100	105	1							
1	96	93	38	43	44	16	58	145	3	116	146	42	23	24	73	41	1
1	14	118	98	101	142	45	120	141	39	15	143	88	138	1			
1	113	148	7	97	119	61	90	1									

Αποτελέσματα για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών, μοιάζει πολύ με την λύση της προηγούμενης παραλλαγής του αλγορίθμου και είναι ελαφρώς βελτιωμένη. Έχει κόστος 927.4242. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (865,94) είναι:

$$\frac{927.4242 - 865{,}94}{865{,}94} = 7{,}10\%$$

Γραφικά η λύση έχει ως εξής:



Η συγκεκριμένη λύση, όπως φαίνεται και από το σχήμα, αποτελείται επίσης από 10 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 10 αυτές διαδρομές έχουν ως εξής:

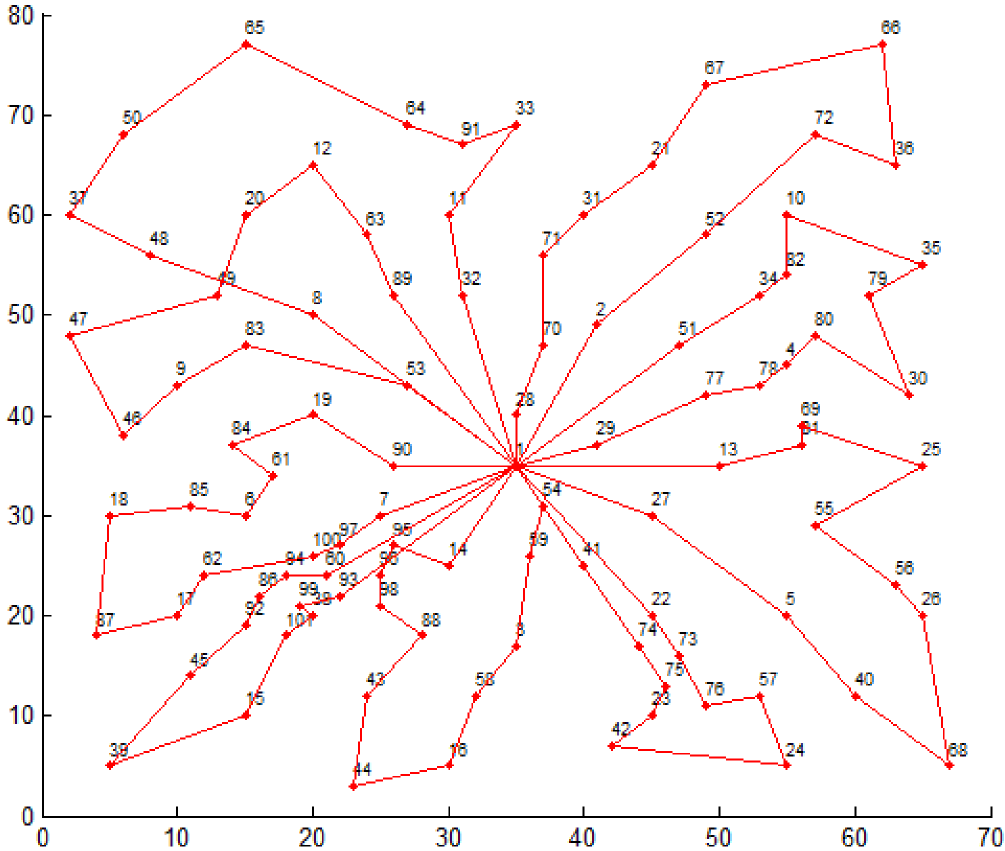
1	29	77	78	4	80	30	25	69	81	13	27	1	
1	19	83	49	48	47	37	50	20	8	53	1		
1	7	97	100	94	86	92	101	99	60	96	95	1	
1	14	88	43	16	44	15	39	45	38	93	98	1	
1	6	85	62	17	87	18	46	9	84	61	90	1	
1	59	3	58	42	23	76	75	73	74	22	41	54	1
1	5	57	24	68	40	26	56	55	1				
1	89	63	12	65	64	91	33	11	32	1			
1	79	35	36	72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

Αποτελέσματα για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών με χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών όπου έγινε χρήση της μεθόδου GRASP έχει κόστος 902.9253. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (865,94) είναι μόλις:

$$\frac{902.9253 - 865{,}94}{902.9253} = 4{,}10\%$$

Γραφικά έχει ως εξής:



Αυτή η λύση αποτελείται από εννέα διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι εννέα αυτές διαδρομές έχουν ως εξής:

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

4.10 Παράδειγμα 9

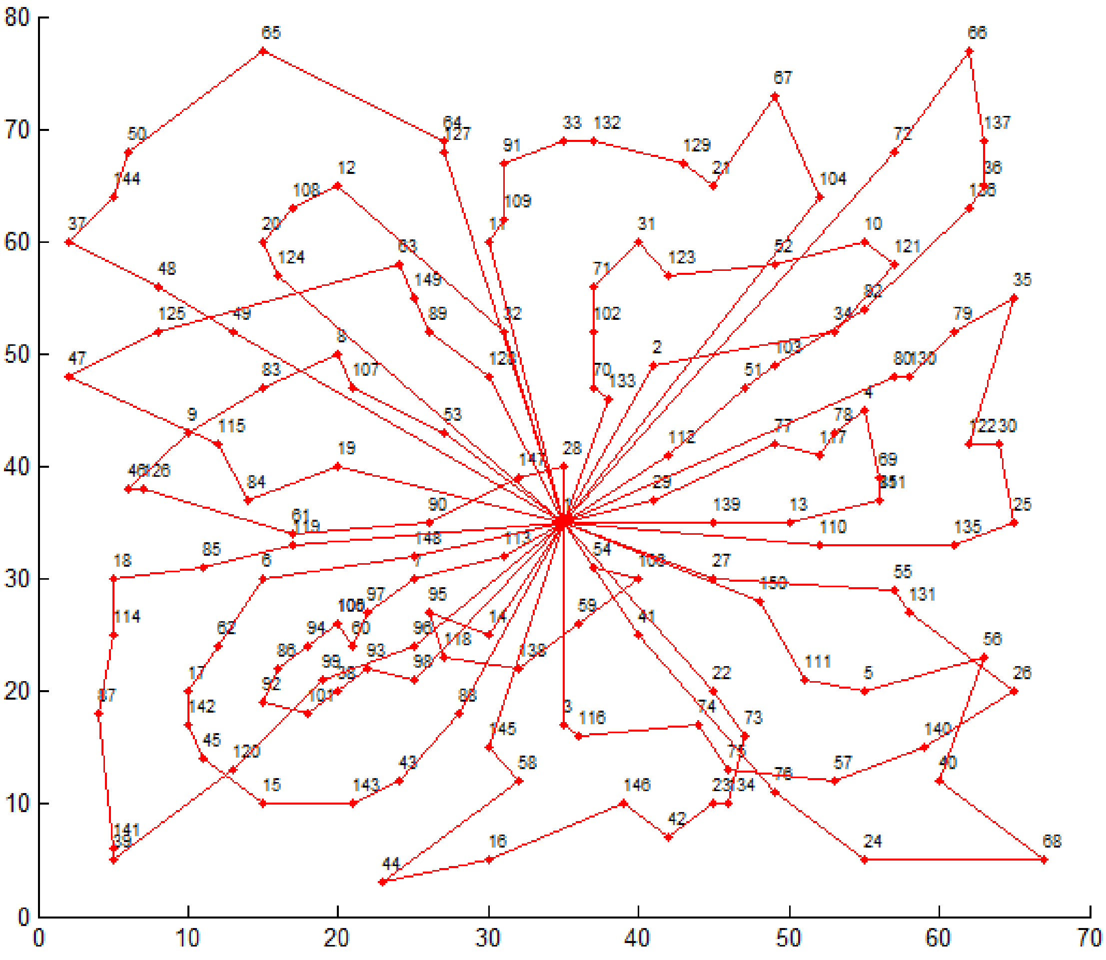
Σε αυτό το παράδειγμα υπάρχουν 76 κόμβοι. Ο κόμβος 1 αντιστοιχεί στην αποθήκη και οι υπόλοιποι 75 στους πελάτες. Κάθε πελάτης έχει συγκεκριμένες συντεταγμένες θέσης και συγκεκριμένη ζήτηση, ακριβή νούμερα για τα οποία δίνονται στο παράρτημα. Θεωρείται ότι όλα τα διαθέσιμα φορτηγά έχουν την ίδια χωρητικότητα, η οποία είναι ίση με 140. Επίσης, καμία διαδρομή δεν πρέπει να έχει συνολικό μήκος πάνω από 160.

<u>Αποτελέσματα για τον Μεμετικό Αλγόριθμο ανεξάρτητα από τη χρήση της μεθόδου GRASP</u>

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο ανεξάρτητα από τη χρήση έχει κόστος 1347,3129. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1162,55) είναι:

$$\frac{1347{,}3129 - 1162{,}55}{1162{,}55} = 15{,}89\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από 16 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 16 αυτές διαδρομές έχουν ως εξής:

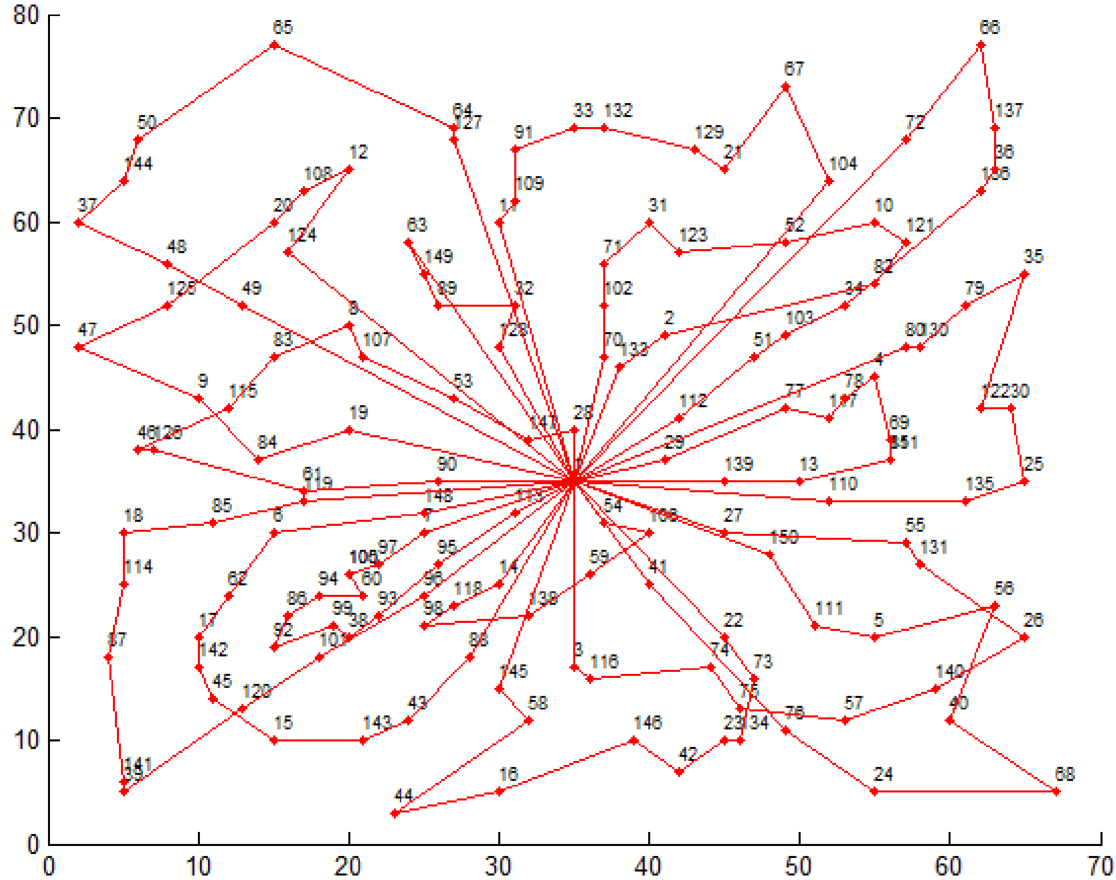
1	28	147	90	61	126	46	9	83	8	107	53	1		
1	19	84	115	47	125	63	149	89	128	1				
1	49	48	37	144	50	65	64	127	1					
1	11	109	91	33	132	129	21	67	104	1				
1	80	130	79	35	122	30	25	135	110	1				
1	119	85	18	114	87	141	39	120	99	96	1			
1	88	43	143	15	45	142	17	62	6	148	1			
1	113	7	97	60	100	105	94	86	92	101	38	93	98	1
1	14	95	118	138	59	106	54	1						
1	72	66	137	36	136	82	103	51	112	1				
1	133	70	102	71	31	123	52	10	121	34	2	1		
1	41	76	24	68	40	56	5	111	150	1				
1	22	73	134	23	42	146	16	44	58	145	1			
1	3	116	74	75	57	140	26	131	55	27	1			
1	29	77	117	78	4	69	81	151	13	139	1			
1	124	20	108	12	32	1								

Αποτελέσματα για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών ανεξάρτητα από τη χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών ανεξάρτητα από τη χρήση της μεθόδου GRASP έχει κόστος 1333,8350. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1162,55) είναι μόλις:

$$\frac{1333{,}8350 - 1162{,}55}{1162{,}55} = 14{,}73\%$$

Γραφικά έχει ως εξής:



Αυτή η λύση αποτελείται από δεκαέξι διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι δεκαέξι αυτές διαδρομές έχουν ως εξής:

1	28	147	53	107	8	83	115	46	126	61	90	1		
1	19	84	9	47	125	20	108	12	124	1				
1	49	48	37	144	50	65	64	127	1					
1	11	109	91	33	132	129	21	67	104	1				
1	133	2	82	121	10	52	123	31	71	102	70	1		
1	7	97	105	100	60	94	86	92	99	38	93	95	113	1
1	148	6	62	17	142	45	15	143	43	88	1			
1	96	101	120	39	141	87	114	18	85	119	1			
1	80	130	79	35	122	30	25	135	110	1				
1	14	118	98	138	59	106	54	1						
1	72	66	137	36	136	34	103	51	112	1				
1	41	76	24	68	40	56	5	111	150	1				
1	22	73	134	23	42	146	16	44	58	145	1			
1	3	116	74	75	57	140	26	131	55	27	1			
1	29	77	117	78	4	69	81	151	13	139	1			
1	63	149	89	32	128	1								

4.11 Παράδειγμα 10

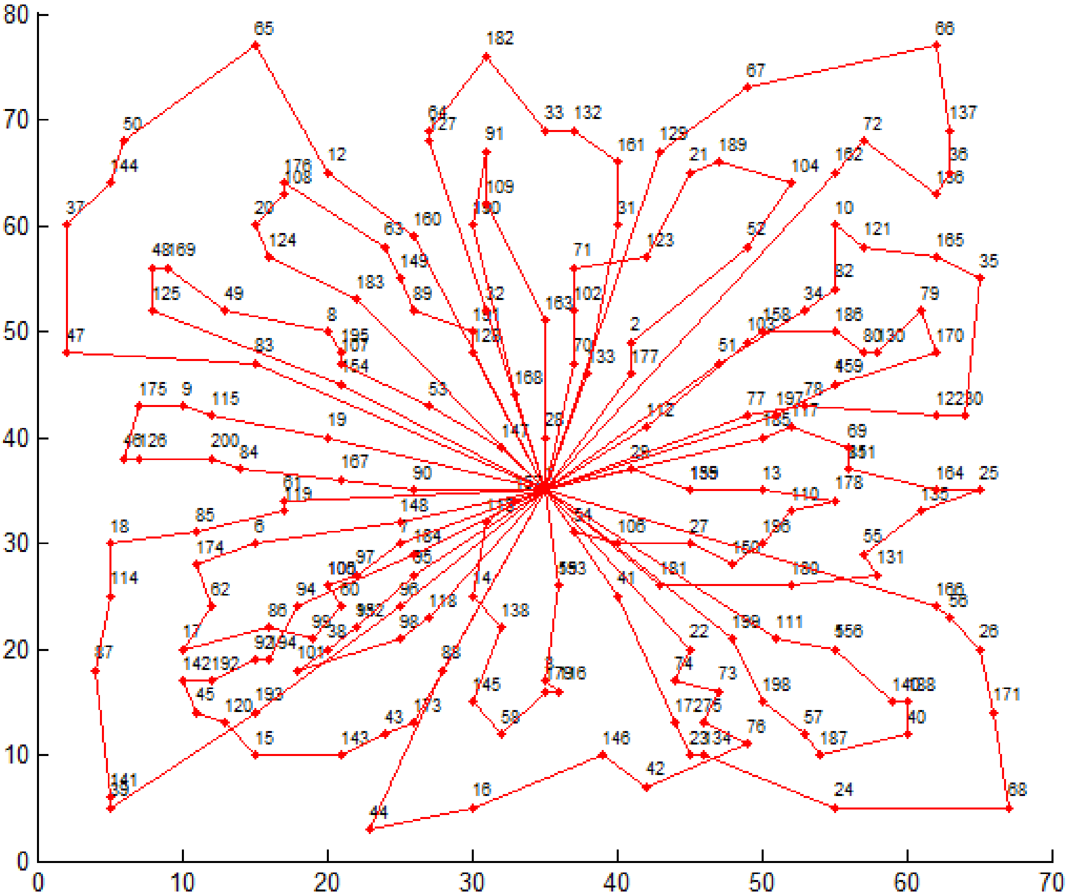
Στο 10^{ο} παράδειγμα υπάρχουν 200 κόμβοι όπου ο κόμβος 1 αντιστοιχεί στην αποθήκη και οι υπόλοιποι 199 στους πελάτες. Κάθε πελάτης έχει συγκεκριμένες συντεταγμένες θέσης και συγκεκριμένη ζήτηση. Οι τιμές για τις συντεταγμένες των κόμβων και τη ζήτησή τους δίνονται στο παράρτημα. Επίσης θεωρείται ότι όλα τα διαθέσιμα φορτηγά έχουν την ίδια χωρητικότητα, η οποία είναι ίση με 200. Τέλος, καμία διαδρομή δεν πρέπει να έχει συνολικό μήκος πάνω από 200 μονάδες.

<u>Αποτελέσματα για τον Μεμετικό Αλγόριθμο ανεξάρτητα από τη χρήση της μεθόδου GRASP</u>

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο ανεξάρτητα από τη χρήση της μεθόδου Grasp, έχει κόστος 1563,8347. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1395,85) είναι:

$$\frac{1563{,}8347 - 1395{,}85}{1395{,}85} = 12{,}03\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από είκοσι διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (αποθήκη). Οι είκοσι αυτές διαδρομές έχουν ως εξής:

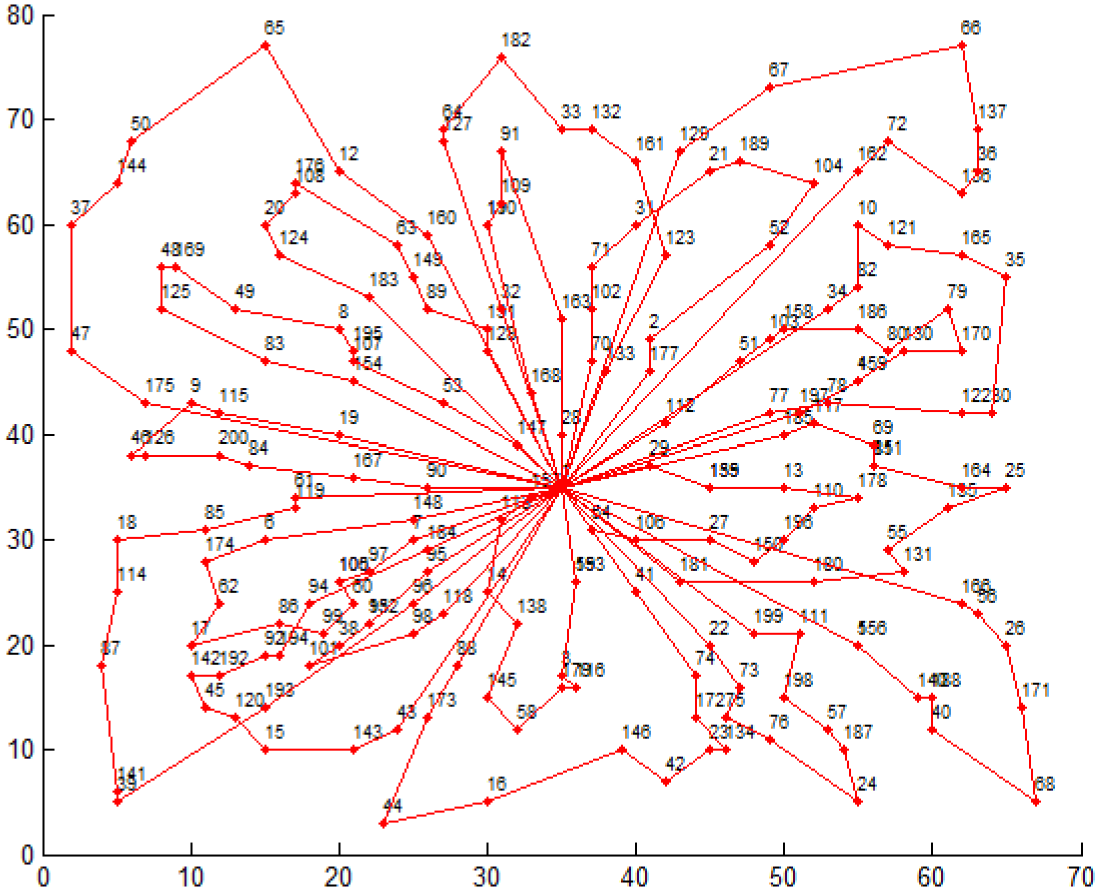
1	54	106	27	150	196	110	178	13	155	139	29	1	
1	181	180	131	55	135	25	164	81	151	69	117	185	1
1	197	159	4	170	79	130	80	186	158	103	112	1	
1	51	34	82	10	121	165	35	30	122	78	77	1	
1	157	113	14	138	145	58	179	116	3	59	153	1	
1	28	163	109	91	11	190	168	1					
1	166	56	26	171	68	24	134	23	172	41	1		
1	199	198	57	187	40	188	140	156	5	111	1		
1	148	6	174	62	17	86	99	60	105	100	97	7	1
1	154	125	48	169	49	8	195	107	53	147	1		
1	61	119	85	18	114	87	141	39	193	96	1		
1	160	12	65	50	144	37	47	83	1				
1	32	127	64	182	33	132	161	31	133	1			
1	22	74	73	75	76	42	146	16	44	88	1		
1	173	43	143	15	120	45	142	192	92	194	94	184	1
1	118	98	101	38	152	93	95	1					
1	162	72	136	36	137	66	67	129	1				
1	183	124	20	108	176	63	149	89	191	128	1		
1	177	2	52	104	189	21	123	71	102	70	1		
1	19	115	9	175	46	126	200	84	167	90	1		

<u>Αποτελέσματα για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών ανεξάρτητα από τη χρήση της μεθόδου GRASP</u>

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών ανεξάρτητα με το αν έγινε χρήση ή όχι της μεθόδου GRASP, είναι βελτιωμένη. Έχει κόστος 1553.3715. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1395,85) είναι:

$$\frac{1553{,}3715 - 1395{,}85}{1395{,}85} = 11{,}28\%$$

Γραφικά η λύση έχει ως εξής:



Η συγκεκριμένη λύση, όπως φαίνεται και από το σχήμα, αποτελείται επίσης από 20 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 20 αυτές διαδρομές έχουν ως εξής:

1	54	106	27	150	196	110	178	13	155	139	29	1	
1	181	180	131	55	135	25	164	81	151	69	117	185	1
1	197	159	4	130	170	79	80	186	158	103	51	112	1
1	34	82	10	121	165	35	30	122	78	77	1		
1	157	113	14	138	145	58	179	116	3	59	153	1	
1	28	163	91	109	190	11	168	1					
1	166	56	26	171	68	40	188	140	156	5	1		
1	148	6	174	62	17	86	99	60	105	100	97	7	1
1	154	83	125	48	169	49	8	195	107	53	147	1	
1	133	123	161	132	33	182	64	127	32	1			
1	22	73	75	76	24	187	57	198	111	199	1		
1	96	193	39	141	87	114	18	85	119	61	1		
1	41	74	172	134	23	42	146	16	44	173	88	1	
1	43	143	15	120	45	142	192	92	194	94	184	1	
1	118	98	101	38	152	93	95	1					
1	162	72	136	36	137	66	67	129	1				
1	175	47	37	144	50	65	12	160	1				
1	128	191	89	149	63	176	108	20	124	183	1		
1	177	2	52	104	189	21	31	71	102	70	1		
1	19	115	9	46	126	200	84	167	90	1			

4.12 Παράδειγμα 11

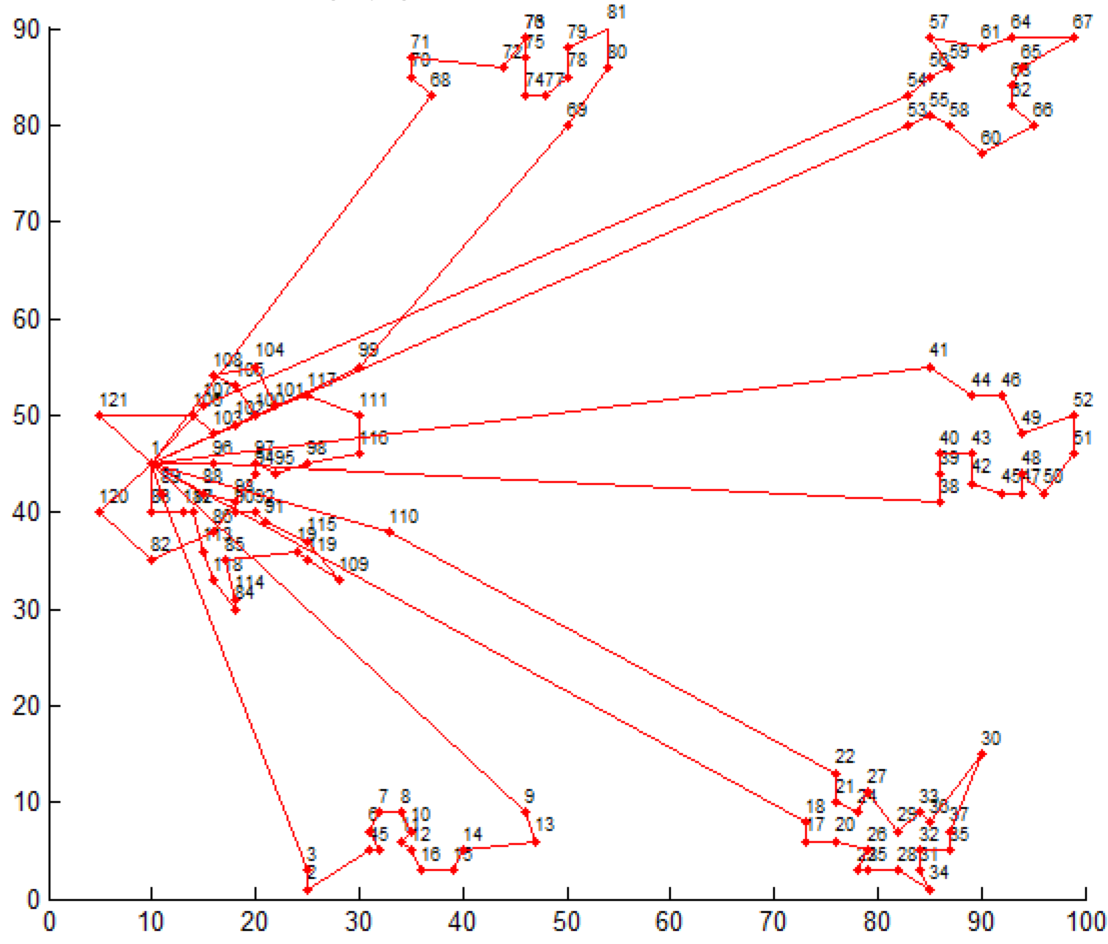
Σε αυτό το παράδειγμα υπάρχουν 121 κόμβοι. Ο κόμβος 1 αντιστοιχεί στην αποθήκη και οι υπόλοιποι 120 στους πελάτες. Κάθε πελάτης έχει συγκεκριμένες συντεταγμένες θέσης και συγκεκριμένη ζήτηση, ακριβή νούμερα για τα οποία δίνονται στο παράρτημα. Θεωρείται ότι όλα τα διαθέσιμα φορτηγά έχουν την ίδια χωρητικότητα, η οποία είναι ίση με 200. Ακόμα, θεωρείται ότι δεν υπάρχουν χρονικοί περιορισμοί.

<u>Αποτελέσματα για τον Μεμετικό Αλγόριθμο ανεξάρτητα από τη χρήση της μεθόδου GRASP</u>

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο ανεξάρτητα με το αν έγινε ή όχι χρήση της μεθόδου Grasp έχει κόστος 1054,7181. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1042,11) είναι:

$$\frac{1054{,}7181 - 1042{,}11}{1042{,}11} = 1{,}21\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από 7 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 7 αυτές διαδρομές έχουν ως εξής:

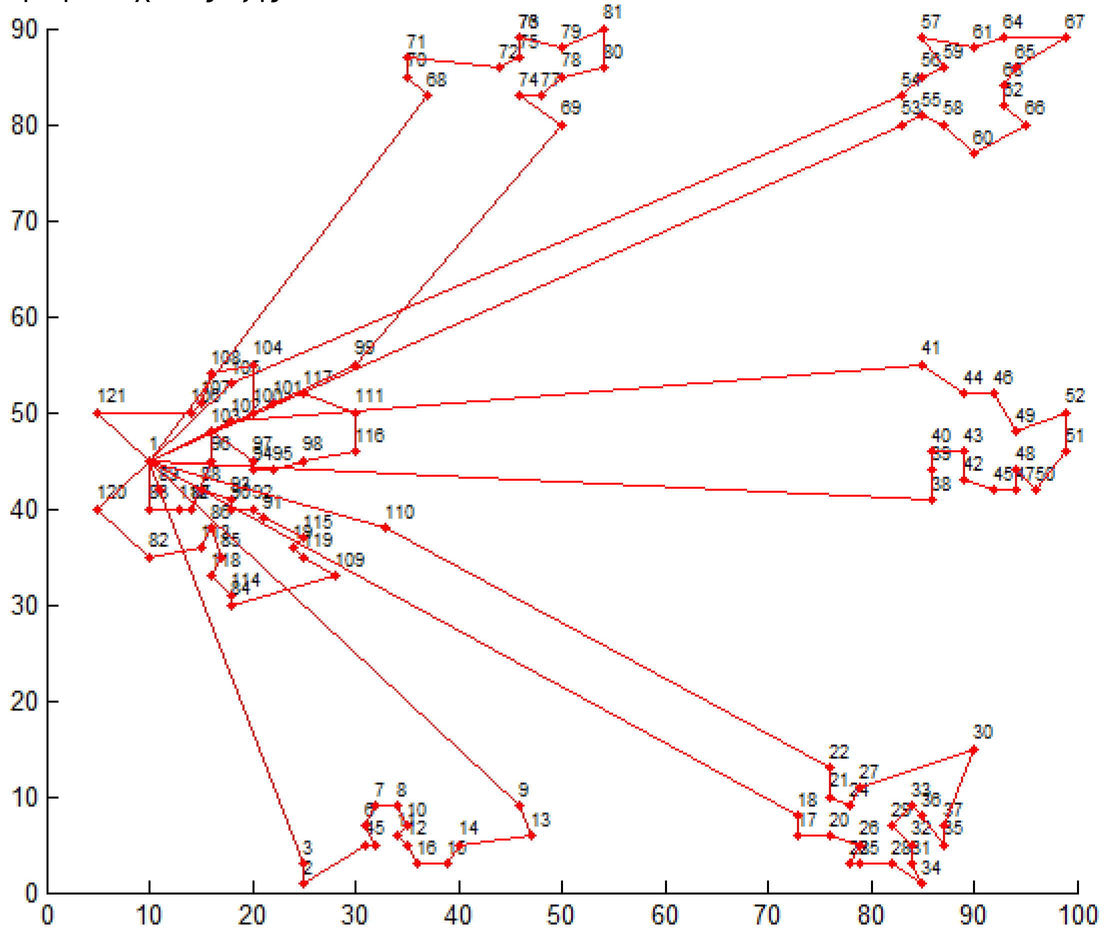
1	110	22	21	24	27	29	33	36	30	37	…	
…	35	32	31	34	28	25	23	26	20	17	18	1
1	9	13	14	15	16	12	11	10	8	7	…	
…	6	5	4	2	3	89	1					
1	96	38	39	40	43	42	45	47	48	50	…	
…	51	52	49	46	44	41	1					
1	68	70	71	72	73	76	75	74	77	78	…	
…	79	81	80	69	99	1						
1	83	112	87	113	118	84	114	85	19	119	…	
…	109	115	91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

<u>Αποτελέσματα για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών ανεξάρτητα από τη χρήση της μεθόδου GRASP</u>

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών ανεξάρτητα από τη χρήση της μεθόδου GRASP έχει κόστος 1048,1585. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1042,11) είναι μόλις:

$$\frac{1048{,}1585 - 1042{,}11}{1042{,}11} = 0{,}58\%$$

Γραφικά έχει ως εξής:



Αυτή η λύση αποτελείται από επτά διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι επτά αυτές διαδρομές έχουν ως εξής:

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

4.13 Παράδειγμα 12

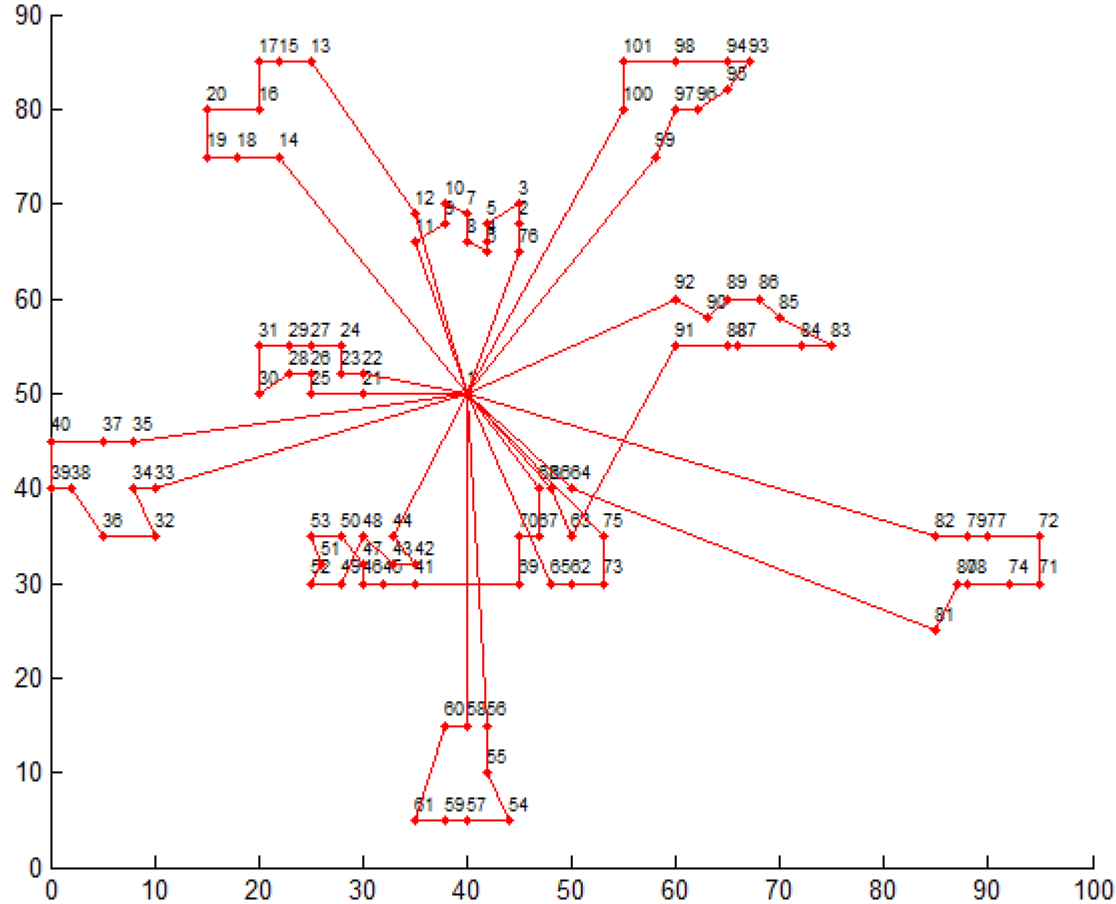
Στο 12ο παράδειγμα υπάρχουν 101 κόμβοι όπου ο κόμβος 1 αντιστοιχεί στην αποθήκη και οι υπόλοιποι 100 στους πελάτες. Κάθε πελάτης έχει συγκεκριμένες συντεταγμένες θέσης και συγκεκριμένη ζήτηση. Οι τιμές για τις συντεταγμένες των κόμβων και τη ζήτησή τους δίνονται στο παράρτημα. Επίσης θεωρείται ότι όλα τα διαθέσιμα φορτηγά έχουν την ίδια χωρητικότητα, η οποία είναι ίση με 200. Τέλος δεν υπάρχουν χρονικοί περιορισμοί.

Αποτελέσματα για τον Μεμετικό Αλγόριθμο ανεξάρτητα από τη χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο ανεξάρτητα από τη χρήση της μεθόδου Grasp, έχει κόστος 862,8681. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (819,56) είναι:

$$\frac{862{,}8681 - 819{,}56}{819{,}56} = 5{,}28\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από 10 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (αποθήκη). Οι 10 αυτές διαδρομές έχουν ως εξής:

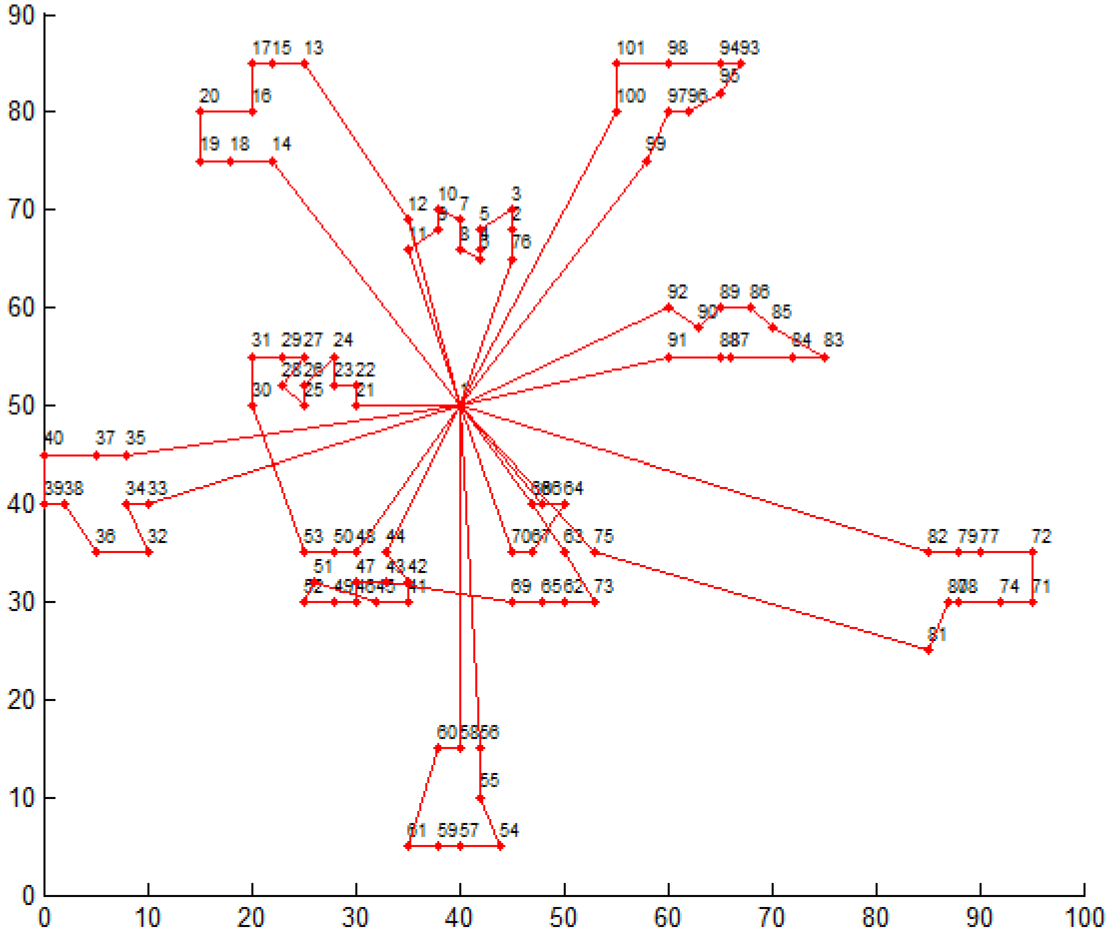
1	68	67	70	69	41	45	46	47	50	53	51	52	…
…	49	48	43	42	44	1							
1	66	63	91	88	87	84	83	85	86	89	90	92	1
1	64	81	80	78	74	71	72	77	79	82	1		
1	14	18	19	20	16	17	15	13	12	1			
1	76	2	3	5	4	6	8	7	10	9	11	1	
1	58	60	61	59	57	54	55	56	1				
1	100	101	98	94	93	95	96	97	99	1			
1	33	34	32	36	38	39	40	37	35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

Αποτελέσματα για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών ανεξάρτητα από τη χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών ανεξάρτητα με το αν έγινε χρήση ή όχι της μεθόδου GRASP, είναι βελτιωμένη. Έχει κόστος 858,6131. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (819,56) είναι:

$$\frac{858{,}6131 - 819{,}56}{819{,}56} = 4{,}77\%$$

Γραφικά η λύση έχει ως εξής:



Η συγκεκριμένη λύση, όπως φαίνεται και από το σχήμα, αποτελείται επίσης από 10 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 10 αυτές διαδρομές έχουν ως εξής:

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

4.14 Παράδειγμα13

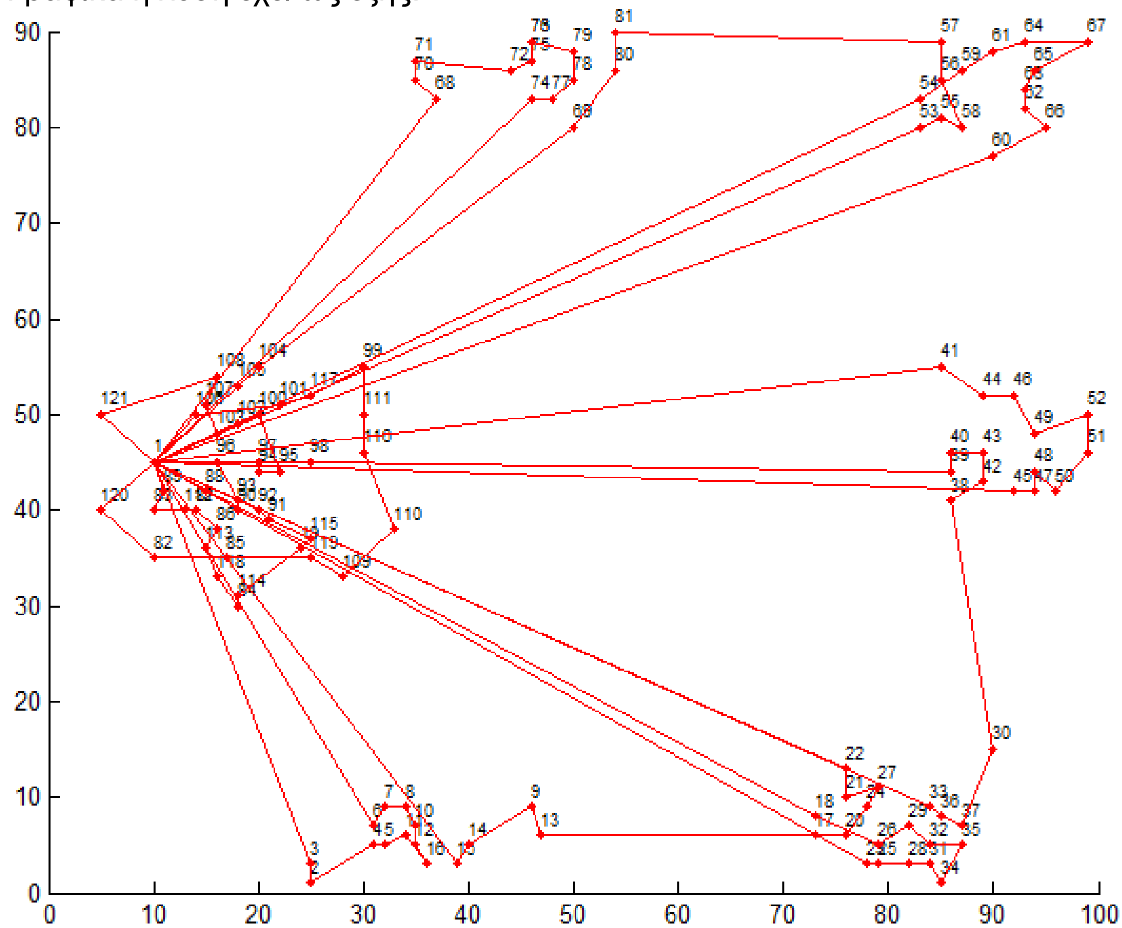
Σε αυτό το παράδειγμα υπάρχουν 121 κόμβοι. Ο κόμβος 1 αντιστοιχεί στην αποθήκη και οι υπόλοιποι 120 στους πελάτες. Κάθε πελάτης έχει συγκεκριμένες συντεταγμένες θέσης και συγκεκριμένη ζήτηση, ακριβή νούμερα για τα οποία δίνονται στο παράρτημα. Θεωρείται ότι όλα τα διαθέσιμα φορτηγά έχουν την ίδια χωρητικότητα, η οποία είναι ίση με 200. Τέλος, καμία διαδρομή δεν πρέπει να έχει συνολικό μήκος πάνω από 720 μονάδες.

Αποτελέσματα για τον Μεμετικό Αλγόριθμο ανεξάρτητα από τη χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο ανεξάρτητα με το αν έγινε ή όχι χρήση της μεθόδου Grasp έχει κόστος 1585,7946. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1541,14) είναι:

$$\frac{1585{,}7946 - 1541{,}14}{1541{,}14} = 2{,}90\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από έντεκα διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι έντεκα αυτές διαδρομές έχουν ως εξής:

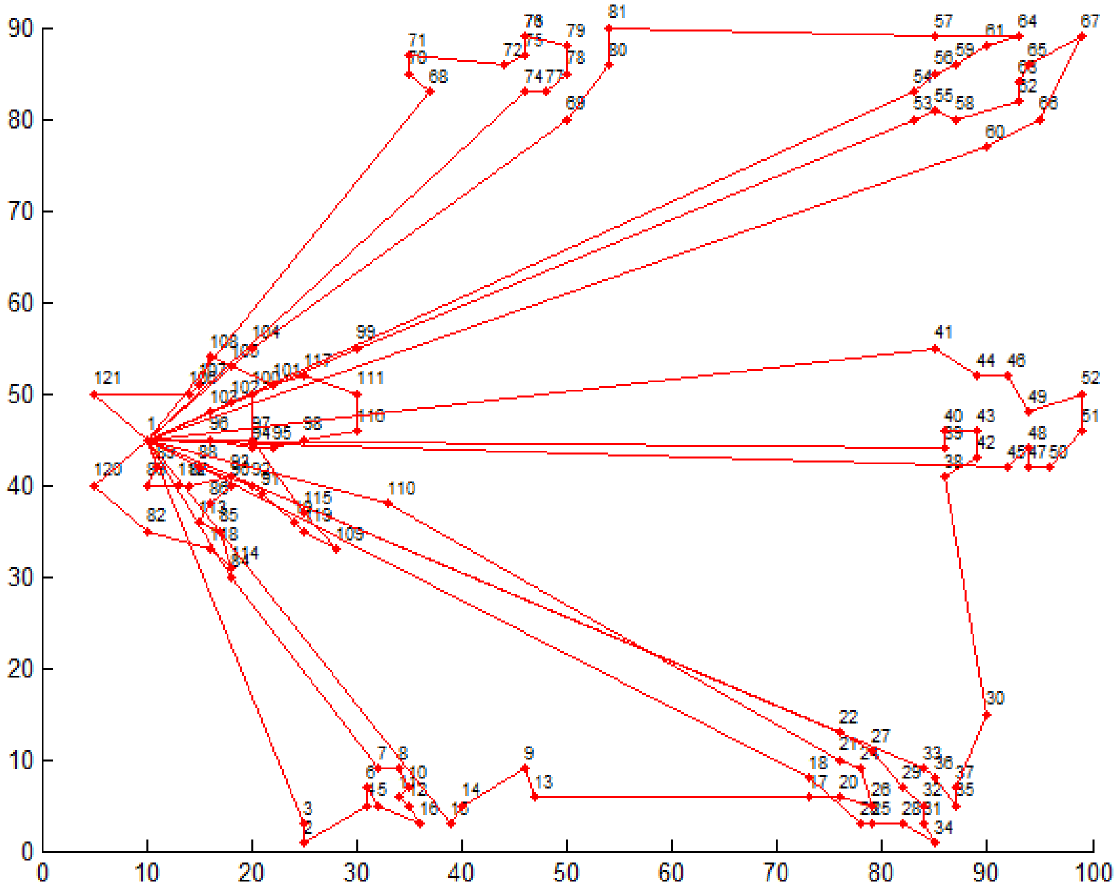
1	89	83	112	87	86	113	118	84	114	19	115	91	92	1
1	88	90	93	96	97	94	95	100	102	103	107	108	121	1
1	106	101	117	99	111	116	110	109	119	85	82	120	1	
1	98	39	40	43	42	38	30	37	36	33	1			
1	22	21	27	24	20	17	13	9	14	15	1			
1	6	7	8	10	12	16	11	5	4	2	3	1		
1	68	70	71	72	75	73	76	79	78	77	74	1		
1	54	59	61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

Αποτελέσματα για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών ανεξάρτητα από τη χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών ανεξάρτητα από τη χρήση της μεθόδου GRASP έχει κόστος 1563,8016. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (1541,14) είναι μόλις:

$$\frac{1563{,}8016 - 1541{,}14}{1541{,}14} = 1{,}47\%$$

Γραφικά έχει ως εξής:



Αυτή η λύση αποτελείται από 11 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι 11 αυτές διαδρομές έχουν ως εξής:

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

4.15 Παράδειγμα 14

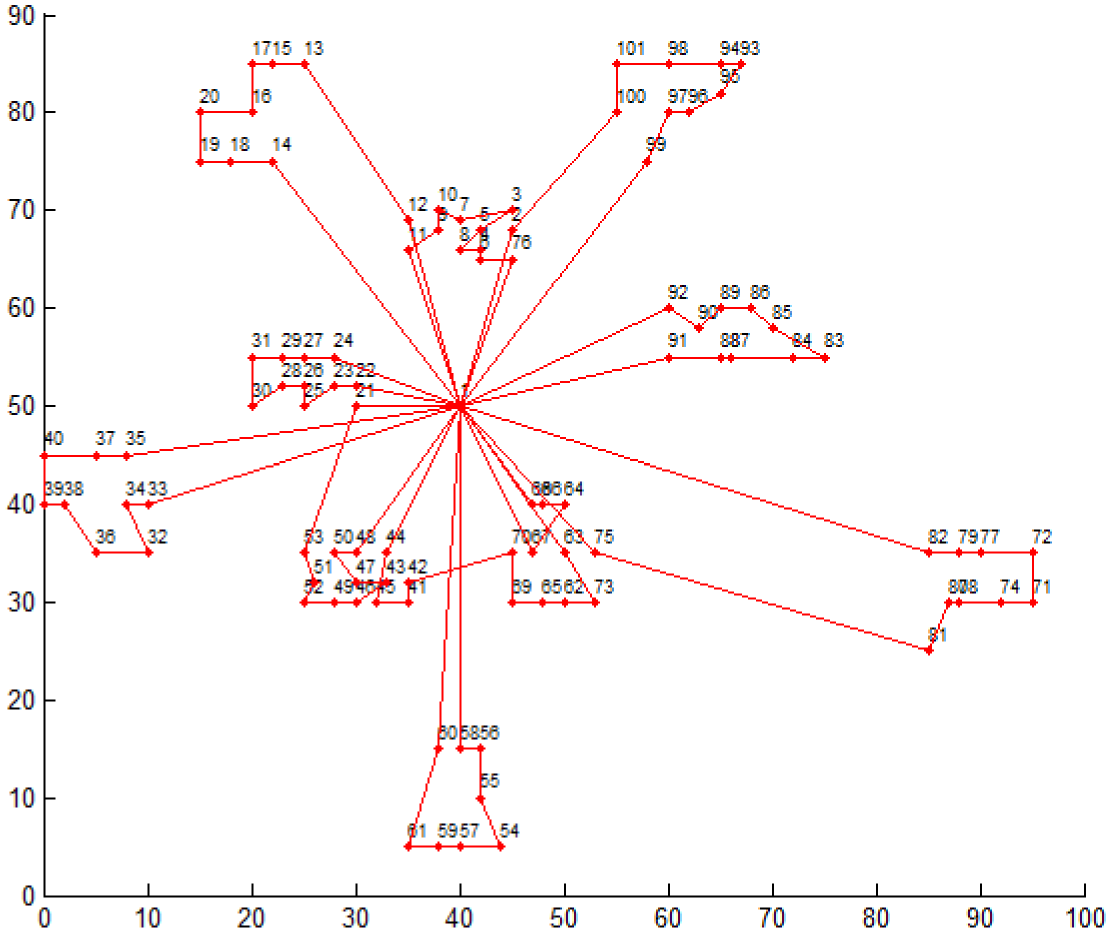
Στο 14° παράδειγμα υπάρχουν 101 κόμβοι όπου ο κόμβος 1 αντιστοιχεί στην αποθήκη και οι υπόλοιποι 100 στους πελάτες. Κάθε πελάτης έχει συγκεκριμένες συντεταγμένες θέσης και συγκεκριμένη ζήτηση. Οι τιμές για τις συντεταγμένες των κόμβων και τη ζήτησή τους δίνονται στο παράρτημα. Επίσης θεωρείται ότι όλα τα διαθέσιμα φορτηγά έχουν την ίδια χωρητικότητα, η οποία είναι ίση με 200. Επίσης, καμία διαδρομή δεν πρέπει να έχει συνολικό μήκος πάνω από 1040.

<u>Αποτελέσματα για τον Μεμετικό Αλγόριθμο ανεξάρτητα από τη χρήση της μεθόδου GRASP</u>

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο (ανεξάρτητα από τη χρήση της μεθόδου Grasp) έχει κόστος 882.7472. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (866,37) είναι:

$$\frac{882.7472 - 866{,}37}{866{,}37} = 1{,}89\%$$

Γραφικά η λύση έχει ως εξής:



Αυτή η λύση αποτελείται από 11 διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (αποθήκη). Οι 11 αυτές διαδρομές έχουν ως εξής:

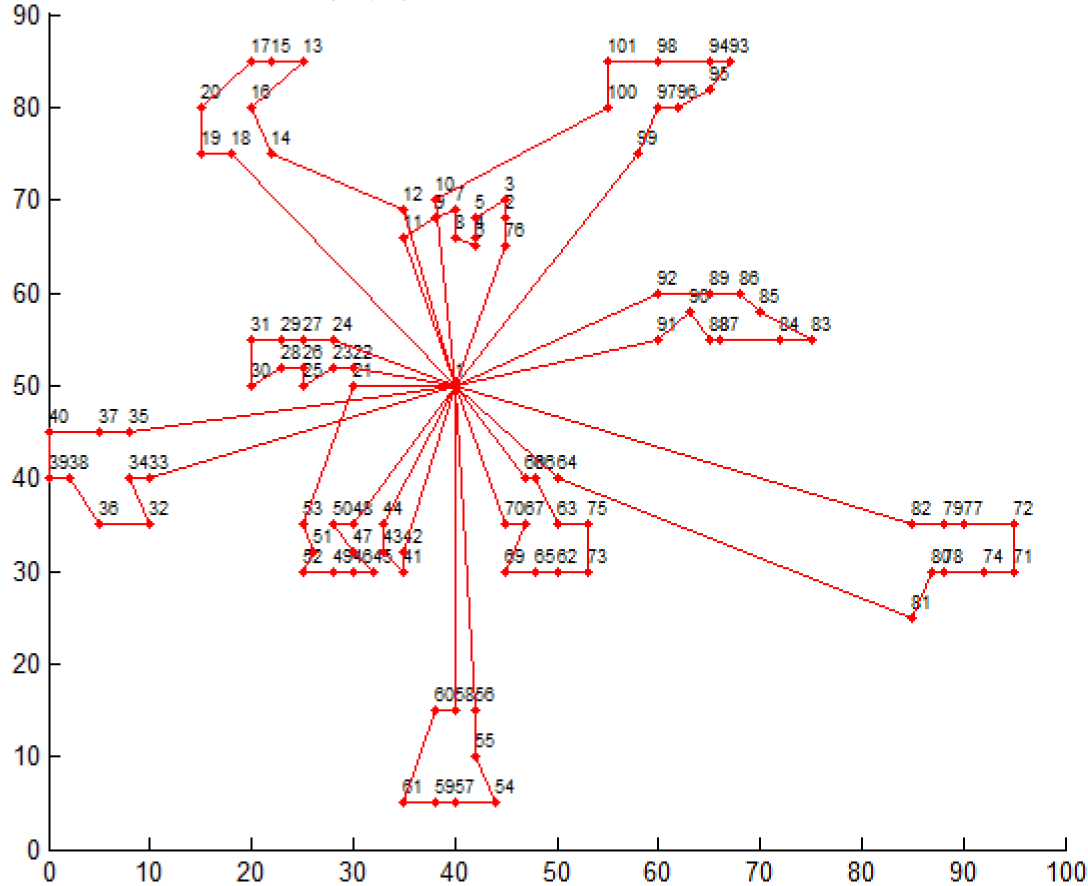
1	91	88	87	84	83	85	86	89	90	92	1
1	82	79	77	72	71	74	78	80	81	75	1
1	63	73	62	65	69	70	42	41	45	44	1
1	22	23	25	26	28	30	31	29	27	24	1
1	21	53	51	52	49	46	43	47	50	48	1
1	14	18	19	20	16	17	15	13	12	1	
1	11	9	10	7	3	5	8	4	6	76	1
1	2	100	101	98	94	93	95	96	97	99	1
1	33	34	32	36	38	39	40	37	35	1	
1	60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

Αποτελέσματα για τον Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών ανεξάρτητα από τη χρήση της μεθόδου GRASP

Η λύση που προέκυψε για τον απλό Μεμετικό Αλγόριθμο Πολλαπλών Πληθυσμών – Νησιών ανεξάρτητα με το αν έγινε χρήση ή όχι της μεθόδου GRASP, είναι βελτιωμένη. Έχει κόστος 876.8419. Η απόκλισή της από το μέχρι στιγμής βέλτιστο αποτέλεσμα από προηγούμενες έρευνες σε παγκόσμια κλίμακα (866,37) είναι:

$$\frac{876.8419 - 866{,}37}{866{,}37} = 1{,}21\%$$

Γραφικά η λύση έχει ως εξής:



Η συγκεκριμένη λύση, όπως φαίνεται και από το σχήμα, αποτελείται επίσης από έντεκα διαδρομές, οι οποίες ξεκινούν και καταλήγουν στον κόμβο 1 (την αποθήκη δηλαδή). Οι έντεκα αυτές διαδρομές έχουν ως εξής:

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

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ

Ο αλγόριθμός μας απέδωσε πάρα πολύ ικανοποιητικά αποτελέσματα. Προγραμματίστηκε σε γλώσσα προγραμματισμού MATLAB, η οποία δεν έχει γρήγορους χρόνους υπολογισμών σε σχέση με τις υπόλοιπες γλώσσες προγραμματισμού. Παρ' όλα αυτά όμως, σε όλα τα παραδείγματα που δοκιμάσαμε τον αλγόριθμο πετύχαμε χαμηλές ποσοστιαίες αποκλίσεις σε σχέση με τα μέχρι στιγμής βέλτιστα αποτελέσματα σε παγκόσμιο επίπεδο. Μάλιστα σε πολλά παραδείγματα οι αποκλίσεις ήταν σχεδόν μηδαμινές και σε δύο από αυτά (το 1^o και το 6^o παράδειγμα) οι αποκλίσεις ήταν ακριβώς 0!

Παρατηρούμε ότι γενικά όταν μεγαλώνει η διάσταση του προβλήματος τα αποτελέσματα χειροτερεύουν. Πράγμα που μας οδηγεί στο συμπέρασμα ότι αν ο αλγόριθμός μας προγραμματιζόταν σε άλλη (ταχύτερη) γλώσσα προγραμματισμού θα απέδιδε αρκετά καλύτερα, πολύ πιθανό να πετυχαίναμε αποτελέσματα ίσα με τα αντίστοιχα μέχρι στιγμής βέλτιστα, σε όλα τα παραδείγματα.

Επίσης κατά τη χρήση του αλγορίθμου διαπιστώσαμε ότι δύο από τα ισχυρότερά του πλεονεκτήματα είναι:

- Το γεγονός ότι δεν δίνουμε μεγάλο βάρος στην τοπική αναζήτηση έτσι ώστε να μην ''εγκλωβιστούμε'' σε τοπικό ελάχιστο. Έτσι με την περιοδική ''μετανάστευση'' των λύσεων αναμιγνύονται στην ίδια ομάδα (νησί) λύσεις που διαφέρουν συνήθως αρκετά μεταξύ τους. Με αποτέλεσμα όταν αυτές συνδυαστούν να μικραίνει η πιθανότητα του να προκύψει μία λύση παρόμοια με τις προηγούμενες. Έτσι ο αλγόριθμος δεν θα συγκλίνει πολύ γρήγορα (δηλαδή δεν θα εγκλωβιστούμε σε τοπικό ελάχιστο). Έτσι πετυχαίνουμε με βεβαιότητα καλύτερα αποτελέσματα.
- Πάρα πολύ επίσης βοηθάει το γεγονός ότι όταν εντοπίσουμε μέσα στην ίδια ομάδα γειτονικές λύσεις που μοιάζουν πολύ τότε διαγράφουμε τις χειρότερες. Αυτή η κίνηση είναι στρατηγικής σημασίας αφού αν δεν διαγραφούν οι λύσεις που μοιάζουν πολύ στις γειτονικές τους, τότε θα συνδυαστούν με αυτές (μέσω της διασταύρωσης) και η πιθανότητα να προκύψει καλύτερη λύση εξαρτάται μονάχα από την διαδικασία της τοπικής αναζήτησης. Έτσι με πάρα πολύ μεγάλη πιθανότητα θα εγκλωβιστούμε γρήγορα σε τοπικό ελάχιστο, πράγμα που αποφεύγουμε όμως επιτυχώς διαγράφοντας τις λύσεις που μοιάζουν με τις υπόλοιπες.

Παράρτημα:

1ο παράδειγμα)

Οι συντεταγμένες και η ζήτηση για κάθε κόμβο έχουν ως εξής:

Κόμβος	Συν. X	Συν. 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

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

2ο παράδειγμα)

Οι συντεταγμένες και η ζήτηση για κάθε κόμβο έχουν ως εξής:

Κόμβος	Συν. X	Συν. Y	Ζήτηση	Κόμβος	Συν. X	Συν. Y	Ζήτηση
1	40	40	0	39	47	66	24
2	22	22	18	40	30	60	16
3	36	26	26	41	30	50	33
4	21	45	11	42	12	17	15
5	45	35	30	43	15	14	11
6	55	20	21	44	16	19	18
7	33	34	19	45	21	48	17
8	50	50	15	46	50	30	21
9	55	45	16	47	51	42	27
10	26	59	29	48	50	15	19
11	40	66	26	49	48	21	20
12	55	65	37	50	12	38	5
13	35	51	16	51	15	56	22
14	62	35	12	52	29	39	12
15	62	57	31	53	54	38	19
16	62	24	8	54	55	57	22
17	21	36	19	55	67	41	16
18	33	44	20	56	10	70	7
19	9	56	13	57	6	25	26
20	62	48	15	58	65	27	14
21	66	14	22	59	40	60	21
22	44	13	28	60	70	64	24
23	26	13	12	61	64	4	13
24	11	28	6	62	36	6	15
25	7	43	27	63	30	20	18
26	17	64	14	64	20	30	11
27	41	46	18	65	15	5	28
28	55	34	17	66	50	70	9
29	35	16	29	67	57	72	37
30	52	26	13	68	45	42	30
31	43	26	22	69	38	33	10
32	31	76	25	70	50	4	8
33	22	53	28	71	66	8	11
34	26	29	27	72	59	5	3
35	50	40	19	73	35	60	1
36	55	50	10	74	27	24	6
37	54	10	12	75	40	20	10
38	60	15	14	76	40	37	20

3ο παράδειγμα)

Οι συντεταγμένες και η ζήτηση για κάθε κόμβο έχουν ως εξής:

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

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

4ο παράδειγμα)

Οι συντεταγμένες και η ζήτηση για κάθε κόμβο έχουν ως εξής:

Κόμβος	Συν. Χ	Συν. Υ	Ζήτηση	Κόμβος	Συν. Χ	Συν. Υ	Ζήτηση
1	35	35	0	39	5	5	16
2	41	49	10	40	60	12	31
3	35	17	7	41	40	25	9
4	55	45	13	42	42	7	5
5	55	20	19	43	24	12	5
6	15	30	26	44	23	3	7
7	25	30	3	45	11	14	18
8	20	50	5	46	6	38	16
9	10	43	9	47	2	48	1
10	55	60	16	48	8	56	27
11	30	60	16	49	13	52	36
12	20	65	12	50	6	68	30
13	50	35	19	51	47	47	13
14	30	25	23	52	49	58	10
15	15	10	20	53	27	43	9
16	30	5	8	54	37	31	14
17	10	20	19	55	57	29	18
18	5	30	2	56	63	23	2
19	20	40	12	57	53	12	6
20	15	60	17	58	32	12	7
21	45	65	9	59	36	26	18
22	45	20	11	60	21	24	28
23	45	10	18	61	17	34	3
24	55	5	29	62	12	24	13
25	65	35	3	63	24	58	19
26	65	20	6	64	27	69	10
27	45	30	17	65	15	77	9
28	35	40	16	66	62	77	20
29	41	37	16	67	49	73	25
30	64	42	9	68	67	5	25
31	40	60	21	69	56	39	36
32	31	52	27	70	37	47	6
33	35	69	23	71	37	56	5
34	53	52	11	72	57	68	15
35	65	55	14	73	47	16	25
36	63	65	8	74	44	17	9
37	2	60	5	75	46	13	8
38	20	20	8	76	49	11	18

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

5ο Παράδειγμα)

Οι συντεταγμένες και η ζήτηση για κάθε κόμβο έχουν ως εξής:

Κόμβος	Συν. X	Συν. Y	Ζήτηση	Κόμβος	Συν. X	Συν. Y	Ζήτηση
1	35	35	0	41	40	25	9
2	41	49	10	42	42	7	5
3	35	17	7	43	24	12	5
4	55	45	13	44	23	3	7
5	55	20	19	45	11	14	18
6	15	30	26	46	6	38	16
7	25	30	3	47	2	48	1
8	20	50	5	48	8	56	27
9	10	43	9	49	13	52	36
10	55	60	16	50	6	68	30
11	30	60	16	51	47	47	13
12	20	65	12	52	49	58	10
13	50	35	19	53	27	43	9
14	30	25	23	54	37	31	14
15	15	10	20	55	57	29	18
16	30	5	8	56	63	23	2
17	10	20	19	57	53	12	6
18	5	30	2	58	32	12	7
19	20	40	12	59	36	26	18
20	15	60	17	60	21	24	28
21	45	65	9	61	17	34	3
22	45	20	11	62	12	24	13
23	45	10	18	63	24	58	19
24	55	5	29	64	27	69	10
25	65	35	3	65	15	77	9
26	65	20	6	66	62	77	20
27	45	30	17	67	49	73	25
28	35	40	16	68	67	5	25
29	41	37	16	69	56	39	36
30	64	42	9	70	37	47	6
31	40	60	21	71	37	56	5
32	31	52	27	72	57	68	15
33	35	69	23	73	47	16	25
34	53	52	11	74	44	17	9
35	65	55	14	75	46	13	8
36	63	65	8	76	49	11	18
37	2	60	5	77	49	42	13
38	20	20	8	78	53	43	14
39	5	5	16	79	61	52	3
40	60	12	31	80	57	48	23

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

Κόμβος	Συν. Χ	Συν. Υ	Ζήτηση
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

6^{ο} παράδειγμα)

Οι συντεταγμένες και η ζήτηση για κάθε κόμβο έχουν ως εξής:

Κόμβος	Συν. Χ	Συν. Υ	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

Κόμβος	Συν. Χ	Συν. Υ	Ζήτηση
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

7^o παράδειγμα)

Οι συντεταγμένες και η ζήτηση για κάθε κόμβο έχουν ως εξής:

Κόμβος	Συν. X	Συν. Y	Ζήτηση	Κόμβος	Συν. X	Συν. Y	Ζήτηση
1	40	40	0	39	47	66	24
2	22	22	18	40	30	60	16
3	36	26	26	41	30	50	33
4	21	45	11	42	12	17	15
5	45	35	30	43	15	14	11
6	55	20	21	44	16	19	18
7	33	34	19	45	21	48	17
8	50	50	15	46	50	30	21
9	55	45	16	47	51	42	27
10	26	59	29	48	50	15	19
11	40	66	26	49	48	21	20
12	55	65	37	50	12	38	5
13	35	51	16	51	15	56	22
14	62	35	12	52	29	39	12
15	62	57	31	53	54	38	19
16	62	24	8	54	55	57	22
17	21	36	19	55	67	41	16
18	33	44	20	56	10	70	7
19	9	56	13	57	6	25	26
20	62	48	15	58	65	27	14
21	66	14	22	59	40	60	21
22	44	13	28	60	70	64	24
23	26	13	12	61	64	4	13
24	11	28	6	62	36	6	15
25	7	43	27	63	30	20	18
26	17	64	14	64	20	30	11
27	41	46	18	65	15	5	28
28	55	34	17	66	50	70	9
29	35	16	29	67	57	72	37
30	52	26	13	68	45	42	30
31	43	26	22	69	38	33	10
32	31	76	25	70	50	4	8
33	22	53	28	71	66	8	11
34	26	29	27	72	59	5	3
35	50	40	19	73	35	60	1
36	55	50	10	74	27	24	6
37	54	10	12	75	40	20	10
38	60	15	14	76	40	37	20

8ο παράδειγμα)

Οι συντεταγμένες και η ζήτηση για κάθε κόμβο έχουν ως εξής:

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

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

9^o παράδειγμα)

Οι συντεταγμένες και η ζήτηση για κάθε κόμβο έχουν ως εξής:

Κόμβος	Συν. Χ	Συν. Υ	Ζήτηση	Κόμβος	Συν. Χ	Συν. Υ	Ζήτηση
1	35	35	0	39	5	5	16
2	41	49	10	40	60	12	31
3	35	17	7	41	40	25	9
4	55	45	13	42	42	7	5
5	55	20	19	43	24	12	5
6	15	30	26	44	23	3	7
7	25	30	3	45	11	14	18
8	20	50	5	46	6	38	16
9	10	43	9	47	2	48	1
10	55	60	16	48	8	56	27
11	30	60	16	49	13	52	36
12	20	65	12	50	6	68	30
13	50	35	19	51	47	47	13
14	30	25	23	52	49	58	10
15	15	10	20	53	27	43	9
16	30	5	8	54	37	31	14
17	10	20	19	55	57	29	18
18	5	30	2	56	63	23	2
19	20	40	12	57	53	12	6
20	15	60	17	58	32	12	7
21	45	65	9	59	36	26	18
22	45	20	11	60	21	24	28
23	45	10	18	61	17	34	3
24	55	5	29	62	12	24	13
25	65	35	3	63	24	58	19
26	65	20	6	64	27	69	10
27	45	30	17	65	15	77	9
28	35	40	16	66	62	77	20
29	41	37	16	67	49	73	25
30	64	42	9	68	67	5	25
31	40	60	21	69	56	39	36
32	31	52	27	70	37	47	6
33	35	69	23	71	37	56	5
34	53	52	11	72	57	68	15
35	65	55	14	73	47	16	25
36	63	65	8	74	44	17	9
37	2	60	5	75	46	13	8
38	20	20	8	76	49	11	18

Κόμβος	Συν. Χ	Συν. Υ	Ζήτηση
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

Κόμβος	Συν. Χ	Συν. Υ	Ζήτηση
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

10° παράδειγμα)

Οι συντεταγμένες και η ζήτηση για κάθε κόμβο έχουν ως εξής:

Κόμβος	Συν. X	Συν. Y	Ζήτηση	Κόμβος	Συν. X	Συν. Y	Ζήτηση
1	35	35	0	41	40	25	9
2	41	49	10	42	42	7	5
3	35	17	7	43	24	12	5
4	55	45	13	44	23	3	7
5	55	20	19	45	11	14	18
6	15	30	26	46	6	38	16
7	25	30	3	47	2	48	1
8	20	50	5	48	8	56	27
9	10	43	9	49	13	52	36
10	55	60	16	50	6	68	30
11	30	60	16	51	47	47	13
12	20	65	12	52	49	58	10
13	50	35	19	53	27	43	9
14	30	25	23	54	37	31	14
15	15	10	20	55	57	29	18
16	30	5	8	56	63	23	2
17	10	20	19	57	53	12	6
18	5	30	2	58	32	12	7
19	20	40	12	59	36	26	18
20	15	60	17	60	21	24	28
21	45	65	9	61	17	34	3
22	45	20	11	62	12	24	13
23	45	10	18	63	24	58	19
24	55	5	29	64	27	69	10
25	65	35	3	65	15	77	9
26	65	20	6	66	62	77	20
27	45	30	17	67	49	73	25
28	35	40	16	68	67	5	25
29	41	37	16	69	56	39	36
30	64	42	9	70	37	47	6
31	40	60	21	71	37	56	5
32	31	52	27	72	57	68	15
33	35	69	23	73	47	16	25
34	53	52	11	74	44	17	9
35	65	55	14	75	46	13	8
36	63	65	8	76	49	11	18
37	2	60	5	77	49	42	13
38	20	20	8	78	53	43	14
39	5	5	16	79	61	52	3
40	60	12	31	80	57	48	23

Κόμβος	Συν. X	Συν. Y	Ζήτηση	Κόμβος	Συν. X	Συν. Y	Ζήτηση
81	56	37	6	121	57	58	28
82	55	54	26	122	62	42	8
83	15	47	16	123	42	57	8
84	14	37	11	124	16	57	16
85	11	31	7	125	8	52	10
86	16	22	41	126	7	38	28
87	4	18	35	127	27	68	7
88	28	18	26	128	30	48	15
89	26	52	9	129	43	67	14
90	26	35	15	130	58	48	6
91	31	67	3	131	58	27	19
92	15	19	1	132	37	69	11
93	22	22	2	133	38	46	12
94	18	24	22	134	46	10	23
95	26	27	27	135	61	33	26
96	25	24	20	136	62	63	17
97	22	27	11	137	63	69	6
98	25	21	12	138	32	22	9
99	19	21	10	139	45	35	15
100	20	26	9	140	59	15	14
101	18	18	17	141	5	6	7
102	37	52	7	142	10	17	27
103	49	49	30	143	21	10	13
104	52	64	16	144	5	64	11
105	20	26	9	145	30	15	16
106	40	30	21	146	39	10	10
107	21	47	15	147	32	39	5
108	17	63	19	148	25	32	25
109	31	62	23	149	25	55	17
110	52	33	11	150	48	28	18
111	51	21	5	151	56	37	10
112	42	41	19	152	22	22	18
113	31	32	29	153	36	26	26
114	5	25	23	154	21	45	11
115	12	42	21	155	45	35	30
116	36	16	10	156	55	20	21
117	52	41	15	157	33	34	19
118	27	23	3	158	50	50	15
119	17	33	41	159	55	45	16
120	13	13	9	160	26	59	29

Κόμβος	Συν. Χ	Συν. Υ	Ζήτηση
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

$11^{ο}$ παράδειγμα)

Οι συντεταγμένες και η ζήτηση για κάθε κόμβο έχουν ως εξής:

Κόμβος	Συν. X	Συν. Y	Ζήτηση	Κόμβος	Συν. X	Συν. Y	Ζήτηση
1	10	45	0	31	84	3	11
2	25	1	25	32	84	5	10
3	25	3	7	33	84	9	3
4	31	5	13	34	85	1	7
5	32	5	6	35	87	5	2
6	31	7	14	36	85	8	4
7	32	9	5	37	87	7	4
8	34	9	11	38	86	41	18
9	46	9	19	39	86	44	14
10	35	7	5	40	86	46	12
11	34	6	15	41	85	55	17
12	35	5	15	42	89	43	20
13	47	6	17	43	89	46	14
14	40	5	13	44	89	52	16
15	39	3	12	45	92	42	10
16	36	3	18	46	92	52	9
17	73	6	13	47	94	42	11
18	73	8	18	48	94	44	7
19	24	36	12	49	94	48	13
20	76	6	17	50	96	42	5
21	76	10	4	51	99	46	4
22	76	13	7	52	99	50	21
23	78	3	12	53	83	80	13
24	78	9	13	54	83	83	11
25	79	3	8	55	85	81	12
26	79	5	16	56	85	85	14
27	79	11	15	57	85	89	10
28	82	3	6	58	87	80	8
29	82	7	5	59	87	86	16
30	90	15	9	60	90	77	19

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

12^ο παράδειγμα)

Οι συντεταγμένες και η ζήτηση για κάθε κόμβο έχουν ως εξής:

Κόμβος	Συν. X	Συν. Y	Ζήτηση	Κόμβος	Συν. X	Συν. Y	Ζήτηση
1	40	50	0	34	8	40	40
2	45	68	10	35	8	45	20
3	45	70	30	36	5	35	10
4	42	66	10	37	5	45	10
5	42	68	10	38	2	40	20
6	42	65	10	39	0	40	30
7	40	69	20	40	0	45	20
8	40	66	20	41	35	30	10
9	38	68	20	42	35	32	10
10	38	70	10	43	33	32	20
11	35	66	10	44	33	35	10
12	35	69	10	45	32	30	10
13	25	85	20	46	30	30	10
14	22	75	30	47	30	32	30
15	22	85	10	48	30	35	10
16	20	80	40	49	28	30	10
17	20	85	40	50	28	35	10
18	18	75	20	51	26	32	10
19	15	75	20	52	25	30	10
20	15	80	10	53	25	35	10
21	30	50	10	54	44	5	20
22	30	52	20	55	42	10	40
23	28	52	20	56	42	15	10
24	28	55	10	57	40	5	30
25	25	50	10	58	40	15	40
26	25	52	40	59	38	5	30
27	25	55	10	60	38	15	10
28	23	52	10	61	35	5	20
29	23	55	20	62	50	30	10
30	20	50	10	63	50	35	20
31	20	55	10	64	50	40	50
32	10	35	20	65	48	30	10
33	10	40	30	66	48	40	10

Κόμβος	Συν. Χ	Συν. Υ	Ζήτηση
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

13ο παράδειγμα)

Οι συντεταγμένες και η ζήτηση για κάθε κόμβο έχουν ως εξής:

Κόμβος	Συν. X	Συν. 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

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

14^{o} παράδειγμα)

Οι συντεταγμένες και η ζήτηση για κάθε κόμβο έχουν ως εξής:

Κόμβος	Συν. X	Συν. Y	Ζήτηση	Κόμβος	Συν. X	Συν. Y	Ζήτηση
1	40	50	0	34	8	40	40
2	45	68	10	35	8	45	20
3	45	70	30	36	5	35	10
4	42	66	10	37	5	45	10
5	42	68	10	38	2	40	20
6	42	65	10	39	0	40	30
7	40	69	20	40	0	45	20
8	40	66	20	41	35	30	10
9	38	68	20	42	35	32	10
10	38	70	10	43	33	32	20
11	35	66	10	44	33	35	10
12	35	69	10	45	32	30	10
13	25	85	20	46	30	30	10
14	22	75	30	47	30	32	30
15	22	85	10	48	30	35	10
16	20	80	40	49	28	30	10
17	20	85	40	50	28	35	10
18	18	75	20	51	26	32	10
19	15	75	20	52	25	30	10
20	15	80	10	53	25	35	10
21	30	50	10	54	44	5	20
22	30	52	20	55	42	10	40
23	28	52	20	56	42	15	10
24	28	55	10	57	40	5	30
25	25	50	10	58	40	15	40
26	25	52	40	59	38	5	30
27	25	55	10	60	38	15	10
28	23	52	10	61	35	5	20
29	23	55	20	62	50	30	10
30	20	50	10	63	50	35	20
31	20	55	10	64	50	40	50
32	10	35	20	65	48	30	10
33	10	40	30	66	48	40	10

Κόμβος	Συν. X	Συν. Y	Ζήτηση
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

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Ιωάννης Μαρινάκης, Αθανάσιος Μυγδαλάς. Σχεδιασμός και Βελτιστοποίηση Εφοδιαστικής Αλυσίδας. Εκδόσεις “σοφία”, Θεσσαλονίκη 2008

[2] Yannis Marinakis - Athanasios Migdalas - Panos M. Pardalos. Multiple phase neighborhood Search—GRASP based on Lagrangean relaxation, random backtracking Lin–Kernighan and path relinking for the TSP. J Comb Optim 17: 134–156, 2007

[3] Ανδρέου Χαράλαμπος. Βελτιστοποίηση Δρομολόγησης Οχημάτων Διανομής των κατεψυγμένων Προϊόντων της Παγκύπριας Εταιρίας Αρτοποιών. (ΔΙΠ) 2005

[4] L.S. Ochi, D.S. Vianna, L.M.A. Drummond, and A.O. Victor. An evolutionary hybrid metaheuristic for solving the vehicle routing problem with heterogeneous fleet. *Lecture notes in computer science*, 1391:187–195, 1998.

[5] L.S. Ochi, D.S. Vianna, L.M.A. Drummond, and A.O. Victor. A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet. *Parallel and Distributed Processing*, 1388:216–224, 1998.

[6] I.H. Osman and S. Salhi. Local search strategies for the vehicle fleet mix problem. In V.J. Rayward-Smith, I.H. Osman, C.R. Reeves, and G.D. Smith, editors, *Modern Heuristic Search Methods*, pages 131–153. Wiley:Chichester, 1996.

[7] M. Gendreau, G. Laporte, C. Musaraganyi, and E.D. Taillard. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 26(12):1153–1173, 1999.

[8] N.A. Wassan and I.H. Osman. Tabu search variants for the mix fleet vehicle routing problem. *Journal of the Operational Research Society*, 53 (7):768–782, 2002.

[9] Y. Rochat and E.D. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 40:147–167, 1995.

[10] C.D. Tarantilis, C.T. Kiranoudis, and V.S. Vassiliadis. A list based threshold accepting metaheuristic for the heterogeneous fixed fleet

vehicle routing problem. *Journal of the Operational Research Society*, 54(1):65–71,2003.

[11] C.D. Tarantilis, C.T. Kiranoudis, and V.S. Vassiliadis. A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *European Journal of Operational Research*, 152(1):148–158, 2004.

[12] F. Li, B.L. Golden, and E.A. Wasil. A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 34:2734–2742, 2007.

[13] B.L. Golden, E.Wasil, J. Kelly, and I.M. Chao. The impact of metaheuristic on solving the vehicle routing problem: algorithms, problem sets, and computational results. In T. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 33–56. Kluwer, Boston, MA, 1998.

[14] M. Dell'Amico, M. Monaci, C. Pagani, and D. Vigo. Heuristic approaches for the fleet size and mix vehicle routing problem with time windows. *Transportation Science*, 2007. To appear.

[15] F.H. Liu and S.Y. Shen. The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research Society*, 50(7): 721–732, 1999.

[16] W. Dullaert, G.K. Janssens, K. Sorensen, and B. Vernimmen. New heuristics for the fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research Society*, 53(11):1232–1238, 2002.

[17] O. Bräysy, W. Dullaert, G. Hasle, D. Mester, and M. Gendreau. An effective multi-restart deterministic annealing methauristic for the fleet size and mix vehicle routing problem with time windows. *Transportation Science*, to appear, 2006.

[18] Roberto Baldacci, Maria Battarra, Daniele Vigo. *Routing a Heterogeneous Fleet of Vehicles* Bruce Golden, S. Raghavan, Edward Wasil, editors, The Vehicle Routing Problem: Latest advances and new challenges, pages 3-28, Springer Science+Business Media: New York, 2008.

[19] *Paolo Toth, Andrea Tramontani*. An Integer Linear Programming Local Search for Capacitated Vehicle Routing Problems*. In* Bruce Golden, S. Raghavan, Edward Wasil, editors, The Vehicle Routing Problem: Latest advances and new challenges, pages 275-296, Springer Science+Business Media: New York, 2008.

[20] J.-F. Cordeau and G. Laporte. A tabu search algorithm for the site dependent vehicle routing problem with time windows. *INFOR*, 39:292–298, 2001.

[21] J.-F. Cordeau, M. Gendreau, and G. Laporte. A tabu search heuristic for periodic and multi–depot vehicle routing problems. *Networks*, 30:105–119, 1997.

[22] D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Comput. Oper. Res.*, 34(8):2403–2435, 2007.