ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Εφαρμογή Μεθευρετικών Αλγορίθμων στον Βέλτιστο Προγραμματισμό της Επιθεώρησης Στάσεων Γραμμών Λεωφορείων

Application of Metaheuristic Algorithms in the Optimal Scheduling of the Bus Line Inspection

Μαργαρίτης Ελευθέριος

Επιβλέποντες:

Κωνσταντίνος Λ. Κεπαπτσόγλου, Επίκουρος Καθηγητής ΕΜΠ

ΑΘΗΝΑ, ΜΑΡΤΙΟΣ 2021

Αφιερωμένη

Στον πατερά μου,

Που του αρέσει να σκαλίζει τα μαθηματικά, την φυσική, τα μηχανήματα,

τις φιλοσοφίες και τις τέχνες!

Κάτι θα βρει να σκαλίσει και εδώ αν πέσει στα χέρια του…

ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά ευχαριστώ τον Κύριο Κωνσταντίνο Κεπαπτσόγλου χωρίς την βοήθεια του οποίου δεν θα είχε ολοκληρωθεί αυτή η εργασία!

Έπειτα οικογένεια…

Έπειτα φίλους…

Έπειτα όλους αυτούς που χτίσανε και χτίζουν τον πολιτισμό στον οποίο είχα την τύχη να γεννηθώ…

ΠΕΡΙΛΗΨΗ

Ο σκοπός αυτής της διπλωματικής εργασίας είναι να βρεθεί μια στρατηγική ώστε να επιθεωρηθεί το δίκτυο του ΟΑΣΑ. Το δίκτυο αυτό αποτελείται από 7962 στάσεις στην περιοχή της Αττικής. Για την επιθεώρηση του πρέπει ένα όχημα να περάσει από όλες τις στάσεις και να τις ελέγξει. Ωστόσο για να γίνει αυτό πρέπει να τεμαχιστεί το δίκτυο σε κομμάτια που η επιθεώρηση τους να μπορεί να ολοκληρωθεί σε ένα λογικό χρόνο! Στην εργασία αυτή προσπαθήσαμε να χωρίσουμε το δίκτυο σε ομάδες (συστάδες) που να μπορούν να επιθεωρηθούν σε περίπου οκτώ ώρες. Δηλαδή πέρα από το πρόβλημα της εύρεσης βέλτιστης δρομολόγησης του οχήματος επιθεώρησης σε κάθε συστάδα υπάρχει και το πρόβλημα ποια είναι και η βέλτιστη στρατηγική ομαδοποίησης. Μελετήθηκε για δυο διαφορετικές περιπτώσεις ταχύτητας και χρόνου επιθεώρησης στάσης, η πρώτη περίπτωση ταχύτητα 10 km/h και χρόνος επιθεώρησης στάσης 1 λεπτό και η δεύτερη περίπτωση ταχύτητα 5 km/h και χρόνος επιθεώρησης στάσης 2 λεπτά. Το πρόβλημα αυτό συναντιέται συχνά στην επιστημονική βιβλιογραφία, αποτελεί μια γενίκευση του προβλήματος του περιπλανώμενου πωλητή (travelling salesman problem) και υπάρχει καλή ομοιότητα απαιτήσεων και δεδομένων του προβλήματος στην ειδική κατηγορία των προβλημάτων δρομολόγησης οχημάτων με χρονικά παράθυρα (VRPTW). Στο πρώτο μέρος για την εύρεση βέλτιστης στρατηγικής ομαδοποίησης θα γίνει χρήση του αλγορίθμου συσταδοποίησης k – means clustering αποκλειστικά και στο δεύτερο μέρος με την βοήθεια του και την χρήση της ανθρώπινης διαίσθησης θα προσπαθήσουμε να έρθουμε σε ακόμα πιο καλά αποτελέσματα. Για την εύρεση βέλτιστης διαδρομής στα πλαίσια της κάθε συστάδας θα εφαρμοστεί ο αλγόριθμος των Lin – Kernighan που θεωρείται από τους πιο αποδοτικούς αλγορίθμους για την επίλυση του κλασσικού προβλήματος του περιπλανώμενου πωλητή. Για την συσταδιοποίηση του δικτύου έγινε χρήση του λογισμικού QGIS και για η εφαρμογή του αλγορίθμου Lin – Kernighan πραγματοποιήθηκε μέσω του λογισμικού Concorde TSP Solver for Windows. Τέλος για ανάλυση, δημιουργία διαγραμμάτων και σύγκριση αποτελεσμάτων έγινε χρήση του Microsoft Excel.

Λέξεις Κλειδιά:

Πρόβλημα περιπλανώμενου πωλητή, Πρόβλημα Δρομολόγησης Οχημάτων, Βελτιστοποίηση, Συσταδιοποίηση, Lin – Kernighan, K – means clustering , QGIS , Concorde TSP Solver

ABSTRACT

In this thesis we will try to tackle a generalized travelling salesman problem. Our problem is that we have to inspect 7962 bus stops in Athens, Greece. We will try to cluster our 7962 geographical points in such a way that given that we will follow a near optimal route in each cluster each inspection we will take close to 8 hours to complete. We will try to solve for two different speed and times spent at each station to complete inspection. First for 10 km/h speed and 1 minute per stop and second 5 km/h and 2 minutes per stop. In order to make good clusters will use extensively the k – means clustering algorithm and for finding near optimal routes in each cluster we will use the Lin – Kernighan heuristic which is considered one of the best heuristics applicable to the travelling salesman problem. For the geographical work and clustering methodology we will use QGIS an effective GIS software and for the Lin – Kernighan implementation the concorde TSP solver for windows an extremely effective solver which is widely available for academic research.

Key Words:

Travelling Salesman Problem (TSP), Vehicle Routing Problem (VRP), Optimization, Heuristic Algorithms, K – means clustering, Lin – Kernighan , QGIS , Concorde Solver

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ → σελ 8
   1.1. Γενικά στοιχεία προβλήματος → σελ 8
   1.2. Στόχος της διπλωματικής εργασίας → σελ 9
   1.3. Δομή της διπλωματικής εργασίας → σελ 10

2. ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ – ΜΑΘΗΜΑΤΙΚΟ ΥΠΟΒΑΘΡΟ ΕΡΓΑΣΙΑΣ → σελ 11
   2.1. Το Πρόβλημα του Περιοδεύοντος Πωλητή (Travelling Salesman Problem- TSP) και προεκτάσεις του → σελ 11
      2.1.1. Το πολλαπλό TSP → σελ 13
   2.2. Πρόβλημα Δρομολόγησης Οχημάτων (Vehicle Routing Problem –VRP) και παραλλαγές του → σελ 14
   2.3. Ευρετικοι αλγόριθμοι και τεχνικές προσπάθειας εύρεσης βέλτιστης λύσης δρομολόγησης → σελ 18
      2.3.1. Τοπική αναζήτηση και αλγόριθμοι ανταλλαγής ζευγών → σελ 20
      2.3.2. Αλγοριθμος των Lin- Kernighan → σελ 22
   2.4. Η ομαδοποίηση Κ μέσων (Κ – means clustering) → σελ 24
   2.5. Η έννοια της ελάχιστης περιβάλλουσας γεωμετρίας (minimum bounding geometry) και η έννοια της χωρικής πυκνότητας σημείων → σελ 26

3. Λογισμικό Εφαρμογής και Λειτουργίες του
   3.1. QGIS → σελ 27
      3.1.1. K – means Clustering → σελ 28
      3.1.2. Minimum Bounding Geometry → σελ 29
   3.2. Concorde TSP Solver For Windows → σελ 32

4. Αντιμετώπιση προβλήματος με δημιουργία επίπεδων συσταδιοποίησης

4.1. Παράμετροι και υποθέσεις του προβλήματος → σελ 40

4.2. Πρώτο επίπεδο ανάλυσης → σελ 41

4.3. Δεύτερο επίπεδο ανάλυσης → σελ 48

4.4. Τρίτο επίπεδο ανάλυσης → σελ 60

4.5. Τέταρτο επίπεδο ανάλυσης – Επιπλέον συσταδιοποιηση για τις ανάγκες της αργής επίλυσης → σελ 72

4.6. Παρουσίαση αποτελεσμάτων και παρατηρήσεις → σελ 93

4.7. Σημαντικά αποτελέσματα → σελ 101

5. Αντιμετώπιση προβλήματος με χρήση της ανθρώπινης διαίσθησης και την βοήθεια του αλγορίθμου K – means και της ελάχιστης περιβάλλουσας γεωμετρίας → σελ 113

5.1. Χρήση αλγορίθμων και δημιουργία υπερσυστάδων κατά επιλογήν → σελ 114

5.2. Τελική διάταξη υπερσυστάδων κατά την γρήγορη και αργή επίλυση → σελ 120

5.3. Αποτελέσματα για κάθε υπερσυστάδα

5.3.1. Αποτελέσματα Concorde TSP Solver για γρήγορη επίλυση → σελ 121

5.3.2. Σύνοψη αποτελεσμάτων υπερσυστάδων γρήγορης επίλυση → σελ 131

5.3.3. Αποτελέσματα Concorde TSP Solver για αργή επίλυση → σελ 132

5.3.4. Σύνοψη αποτελεσμάτων υπερσυστάδων αργής επίλυσης → σελ 151

6. Συμπεράσματα και προτάσεις για περαιτέρω έρευνα

6.1. Μηχανική και επιλεκτική συσταδιοποίηση για την γρήγορη επίλυση → σελ 153

6.2. Μηχανική και επιλεκτική συσταδιοποίηση για την αργή επίλυση → σελ 155

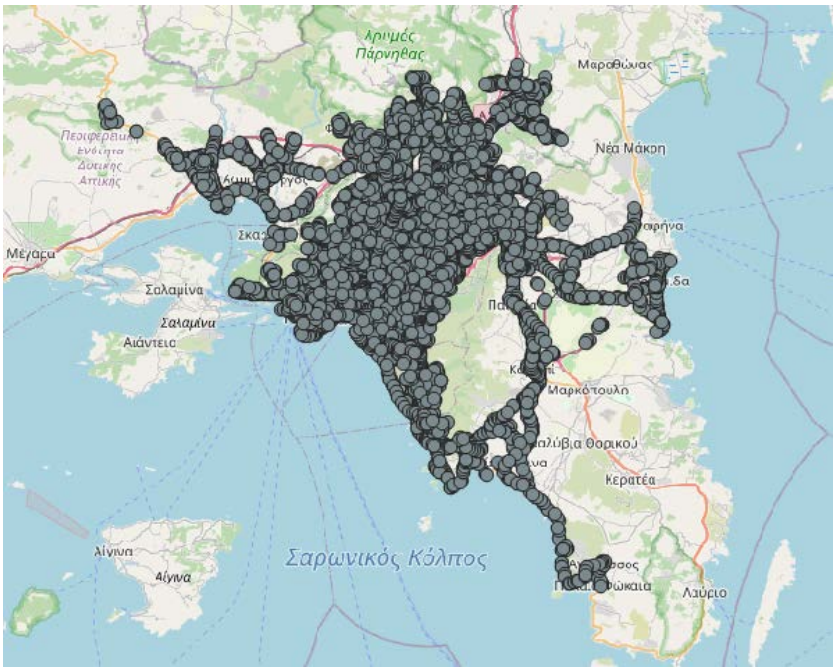
6.3. Προτάσεις για περαιτέρω ερευνά → σελ 157

7. Βιβλιογραφία → σελ 159

1. Εισαγωγή

1.1 Γενικά στοιχεία προβλήματος

Το δίκτυο των γραμμών λεωφορείων και τρόλεϊ στον νομό Αττικής του Οργανισμού Αστικών Συγκοινωνιών Αθηνών (ΟΑΣΑ) περιέχει 7962 διαφορετικές στάσεις. Είναι αναγκαία ανά τακτά χρονικά διαστήματα η επιθεώρηση αυτών των στάσεων για ελέγχους σε φθορές των διάφορων αντικειμένων που έχουν ώστε να είναι λειτουργικές, παγκάκια και στέγαστρα αλλά κυρίως οι πίνακες δρομολογίων και τα χρονοδιαγράμματα αλλά και οι τηλεματική πίνακες στις πιο σύγχρονες στάσεις. Ωστόσο λόγω του μεγάλου πλήθους των στάσεων η διαδικασία της επιθεώρησης γίνεται ιδιαίτερα δύσκολη και χρονοβόρα και αποκτά πολύ μεγάλη σημασία η στρατηγική με την οποία θα επιθεωρηθούν οι στάσεις. Η διαδικασία της επιθεώρησης γίνεται από έναν επιθεωρητή ο οποίος με το όχημα εργασίας περνάει από ένα πλήθος στάσεων και σημειώνει τις παρατηρήσεις του. Το πόσες στάσεις, ανά εργάσιμη ημέρα, και το με ποια σειρά θα επιθεωρηθούν αποτελεί το βασικό πρόβλημα αυτής της εργασίας. Για την ταχύτητα του οχήματος και τον χρόνο επιθεώρησης κάθε στάσης έγιναν δυο υποθέσεις, ταχύτητα 10km/h και χρόνος επιθεώρησης 1 λεπτό (γρήγορη επίλυση) και ταχύτητα 5km/h και χρόνος επιθεώρησης 2 λεπτά (αργή επίλυση).



Σχήμα 1.1 Χάρτης των στάσεων

1.2 Στόχος της διπλωματικής εργασίας

Σκοπός οπότε αυτής της διπλωματικής εργασίας είναι να βρεθεί η βέλτιστη στρατηγική με την οποία θα επιθεωρηθούν οι στάσεις.

Στα μαθηματικά , την επιστήμη των υπολογιστών και τα οικονομικά , ένα πρόβλημα βελτιστοποίησης είναι ένα πρόβλημα εύρεσης της καλύτερης λύσης από όλες τις εφικτές λύσεις . Στόχος της βελτιστοποίησης είναι η εξεύρεση ενός συνδυασμού ανεξάρτητων μεταβλητών που λαμβάνουν πραγματικές ή ακέραιες τιμές και οι οποίες ονομάζονται παράμετροι ή μεταβλητές σχεδιασμού, έτσι ώστε να βελτιστοποιηθεί η αντικειμενική συνάρτηση του προβλήματος. Τα προβλήματα βελτιστοποίησης συνήθως υπόκεινται σε περιορισμούς οι οποίοι αποτελούν και τα φυσικά όρια που τίθενται στην επίτευξη του στόχου. (πηγή: βιβλιογραφία[1])

Το ένα μέρος του προβλήματος που αντιμετωπίζεται σε αυτή την διπλωματική εργασία είναι ένα πρόβλημα βέλτιστης δρομολόγησης. Το πρόβλημα αυτό, εμφανίζεται σε πολλές μορφές, ανάλογα με το πόσο περίπλοκο είναι το περιβάλλον από το οποιο αντλούμε τα δεδομένα μας και τους στόχους μας, και επιλύεται με διάφορους τρόπους που περιλαμβάνουν πλήθος αλγορίθμων βελτιστοποίησης ακέραιου προγραμματισμού. Οι αλγόριθμοι αυτοί προτείνουν βέλτιστες λύσεις δρομολόγησης ή λύσεις που πλησιάζουν τη βέλτιστη δρομολόγηση, λαμβάνοντας υπόψη πολλαπλά κριτήρια και περιορισμούς. Το πρόβλημα της βέλτιστης δρομολόγησης εφαρμόζεται εντός των πλαισίων των στάσεων που περιέχει η εκάστοτε ομάδα που πρέπει να επιθεωρηθεί.

Το άλλο μέρος του προβλήματος είναι η βέλτιστη ομαδοποίηση του δικτύου του ΟΑΣΑ, ώστε η επιτήρηση κάθε ομάδας στάσεων (συστάδα) να μπορεί να επιτηρηθεί σε έναν λογικό χρόνο. Ως βέλτιστος χρόνος ολοκλήρωσης της επιθεώρησης της ομάδας των στάσεων θα θεωρηθούν οι οκτώ ώρες.

Τα δυο αυτά προβλήματα δεν είναι ανεξάρτητα το ένα από το άλλο, καθώς η εύρεση βέλτιστης δρομολόγησης μειώνει τον χρόνο που απαιτεί ολικά η συστάδα για επιθεώρηση και οπότε επηρεάζει το βέλτιστο μέγεθος της, ισχύει και το αντίστροφο. Το γεγονος αυτό

καθιστά ιδιαίτερα δύσκολη την εύρεση βέλτιστης ολικής στρατηγικής επιθεώρησης, η οποία θα μελετηθεί εκτενώς σε αυτή την διπλωματική εργασία.

1.3 Δομή της διπλωματικής εργασίας

Το τεύχος της διπλωματικής εργασίας δομείται ως εξής:

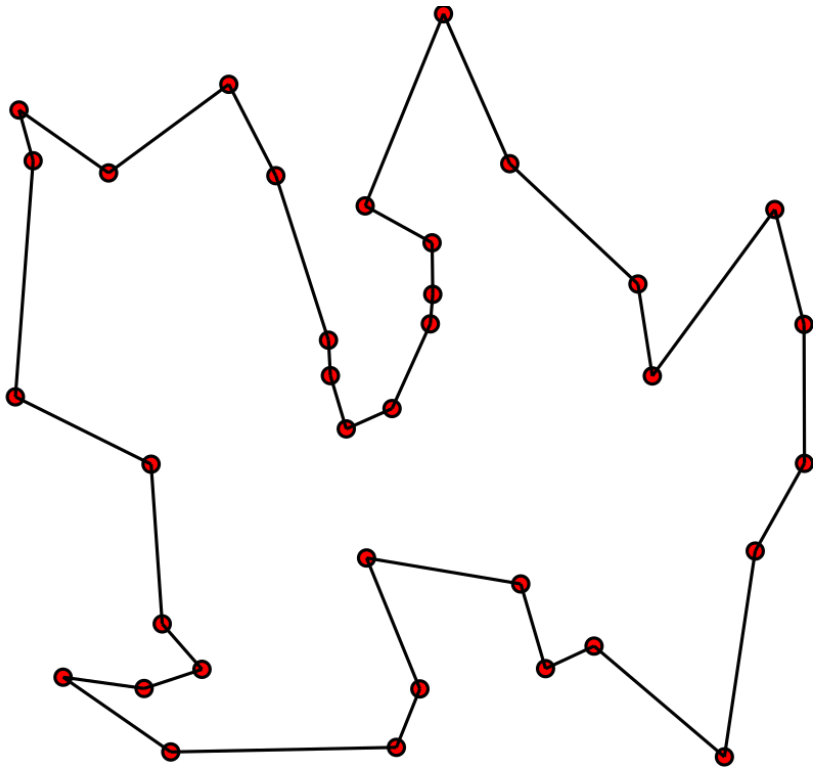
- Στο 2ο κεφάλαιο γίνεται μια βιβλιογραφική ανασκόπηση στο μαθηματικό υπόβαθρο του προβλήματος του πλανοδίου πωλητή (Travelling Salesman Problem) και του προβλήματος δρομολόγησης οχημάτων (Vehicle Routing Problem) καθώς και μερικών παραλλαγών αυτών των προβλημάτων. Έπειτα ακολουθεί η βιβλιογραφική ανασκόπηση διάφορων αλγορίθμων και τεχνικών επίλυσης και βελτιστοποίησης αυτών των προβλημάτων. Τέλος αναλύεται ο αλγόριθμος συσταδιοποίησης Κ μέσων (k – means clustering) και η έννοια της ελάχιστης περιβάλλουσας γεωμετρίας που θα χρησιμοποιηθούν για την κατάλληλη συσταδιοποίηση του δικτύου.
- Στο 3ο κεφάλαιο παρουσιάζονται οι λειτουργίες των λογισμικών QGIS και Concorde TSP Solver που χρησιμοποιήθηκαν για την αντιμετώπιση του προβλήματος.
- Στο 4ο κεφάλαιο γίνεται η πρώτη προσπάθεια εύρεσης βέλτιστης στρατηγικής, κατά την οποία το δίκτυο υποδιαιρείται διαδοχικά σε συστάδες με χρήση του αλγορίθμου k-means, βρίσκεται η βέλτιστη διαδρομή και ο χρόνος επιθεώρησης για κάθε μια και ανάλογα το πόσο υπερβαίνει τις οκτώ ώρες γίνεται επιπλέον συσταδιοποίηση (μηχανική συσταδιοποίηση). Έπειτα ακολουθεί ανάλυση των αποτελεσμάτων.
- Στο 5ο κεφάλαιο χωρίζουμε το δίκτυο σε 200 συστάδες για την αργή επίλυση και σε 500 για την γρήγορη και με την γνώση που προέκυψε από τα αποτελέσματα του 4ου κεφαλαίου προσπαθούμε να δημιουργήσουμε υπερσυστάδες (ένωση συστάδων) που προβλέπουμε ότι η επιθεώρηση τους θα ολοκληρώνεται σε 8 ώρες (επιλεκτική συσταδιοποίηση).
- Στο 6ο κεφάλαιο γίνεται σύγκριση των αποτελεσμάτων επισημαίνονται τα κυριότερα συμπεράσματα και διατυπώνονται ορισμένες προτάσεις για περαιτέρω έρευνα.

2. ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ – ΜΑΘΗΜΑΤΙΚΟ ΥΠΟΒΑΘΡΟ ΕΡΓΑΣΙΑΣ

2.1 Το Πρόβλημα του Περιοδεύοντος Πωλητή (Travelling Salesman Problem- TSP) και προεκτάσεις του

Το Πρόβλημα του Περιοδεύοντος Πωλητή είναι το πλέον διαδεδομένο πρόβλημα δρομολόγησης και παράλληλα ένα από τα πιο δύσκολα σε ότι αφορά στην επίλυση του, το οποίο θέτει την εξής ερώτηση: "Δεδομένης μιας λίστας πόλεων και των αποστάσεων μεταξύ τους, ποια είναι η συντομότερη δυνατή διαδρομή που επισκέπτεται κάθε πόλη ακριβώς μία φορά και επιστρέφει στην πόλη προέλευσης; ". Είναι ένα ιδιαίτερα δύσκολο πρόβλημα πολυπλοκότητας NP-hardness ιδιαίτερα σημαντικό για την θεωρητική επιστήμη των υπολογιστών αλλά και την επιχειρησιακή και λειτουργική ερευνά. (πηγή: βιβλιογραφία[2])

Εδώ παρουσιάζεται η γραφική απεικόνιση της λύσης ενός προβλήματος TSP:



Σχήμα 2.1 γραφική απεικόνιση βέλτιστης λύσης

Τα πρώτα επιστημονικά βήματα σε τέτοιου είδους προβλημάτων βελτιστοποίησης εντοπίστηκαν σε εγχειρίδια για περιοδεύοντες πωλητές το 1832, για περιηγήσεις στην Γερμάνια και την Ελβετία, χωρίς όμως να υπάρχει ιδιαίτερη μαθηματική ανάλυση.

Στον χώρο της Μαθηματικής βελτιστοποίησης μπορεί να ορισθεί ως εξής:

Έστω n πόλεις (στάσεις δικτύου ΟΑΣΑ) με γνωστό κόστος μετακίνησης c_{ij}, από τη μία πόλη i προς μια άλλη j. Ένας πωλητής (όχημα) ξεκινάει από μια πόλη (αρχικός κόμβος του δικτύου) και σκοπός του είναι να επισκεφθεί κάθε πόλη του δικτύου μία και μόνο φορά. Ζητούμενο είναι να βρεθεί η βέλτιστη διαδρομή, δηλαδή η διαδρομή με το λιγότερο δυνατό συνολικό κόστος.

Ορίζονται οι δυαδικές μεταβλητές:

$$x_{ij} = \begin{cases} 1, & \text{εάν το όχημα κάνει χρήση του συνδέσμου} \{i, j\} \\ 0, & \text{αλλιώς} \, \forall i, \ \forall j, \ i \neq j \end{cases}$$

Το μαθηματικό πρότυπο του προβλήματος διαμορφώνεται ως εξής:

$$\min \quad \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} c_{ij} x_{ij}$$

Με τους περιορισμούς:

$$\sum_{j=1}^{n} x_{ij} = 1, \ i = 1, ..., n$$

$$\sum_{i=1}^{n} x_{ij} = 1, \ j = 1, ..., n,$$

$$\sum_{i \in \mathcal{C}} \sum_{j \in \bar{\mathcal{C}}} x_{ij} \geq 1, \ \forall \mathcal{C} \subset \{1, ..., n\}, \ \mathcal{C} \neq \emptyset$$

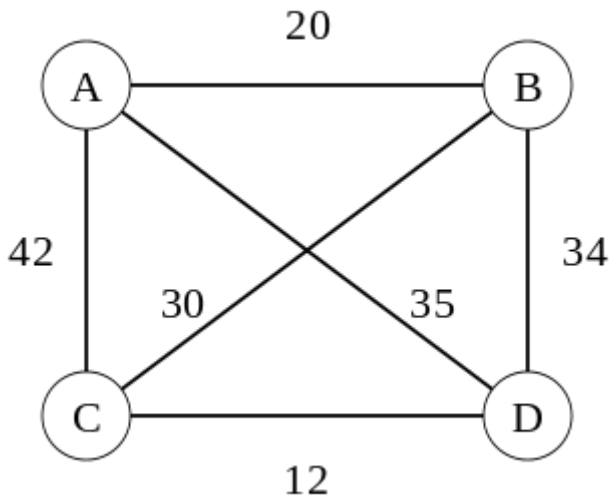
$$x_{ij} \in \{0, 1\}, \ \forall i, \ \forall j, \ i \neq j$$

(πηγή: βιβλιογραφία[3])

Οι περιορισμοί αυτοί βεβαιώνουν ότι κάθε πόλη (πελάτης) δέχεται επίσκεψη μία μόνο φορά.

Ωστόσο το πρόβλημα μπορεί να μοντελοποιηθεί και με μη κατευθυνόμενη γραφική μορφή. Οι πόλεις είναι τα ονοματισμένα σημεία (κύκλοι στο γράφημα) οι διαδρομές είναι τα ευθύγραμμα τμήματα που τα ενώνουν και η απόσταση μιας διαδρομής είναι το βάρος που αναγράφεται σε κάθε ευθύγραμμο τμήμα.

Γραφική απεικόνιση συμμετρικού προβλήματος TSP τεσσάρων πόλεων:



Σχήμα 2.2 γραφική απεικόνιση προβλήματος

Η γενική μορφή του μελετήθηκε το 1930 από τον Karl Menger ο οποίος καθόρισε και το πρόβλημα το οποίο αποτελεί βασικό αντικείμενο μελέτης και ανάλυσης για πολλά προβλήματα βελτιστοποίησης. (πηγή: βιβλιογραφία[2])

2.1.1 Το πολλαπλό TSP

Μια παραλλαγή του προβλήματος που φαίνεται ιδιαίτερα ενδιαφέρουσα συναντάτε στην επιστημονική βιβλιογραφία και έχει άμεση σχέση με το δικό μας πρόβλημα είναι το λεγόμενο πολλαπλό πρόβλημα του πλανοδίου πωλητή(multiple – TSP)

Αντί για ένα μόνο πωλητή έχουμε m πωλητές διαθέσιμους οι οποίοι βρίσκονται όλοι στην πόλη n + 1 και πρέπει να επισκεφτούν τις πόλεις 1,2,…,n . Το κόστος της λύσης είναι η

συνολική απόσταση που διανύεται από το σύνολο των πωλητών (όλοι πρέπει να ταξιδέψουν). Αυτή είναι η βασική περίπτωση όταν στην δρομολόγηση οχημάτων m οχήματα, βρίσκονται σε ένα κοινό σταθμό και πρέπει να εξυπηρετήσουν n πελάτες. (πηγή: βιβλιογραφία[4])

2.2 Πρόβλημα Δρομολόγησης Οχημάτων (Vehicle Routing Problem –VRP) και παραλλαγές του

Στην κλασσική επιστημονική βιβλιογραφία εκτός από το πρόβλημα του πλανοδίου πωλητή πολύ συχνά συναντάτε και το πρόβλημα της βέλτιστης δρομολόγησης οχήματος (vehicle routing problem - VRP), το οποίο στην ουσία αποτελεί μια γενίκευση του προβλήματος. Στο TSP δεδομένα είναι ένα σύνολο πόλεων (ή σημείων) και στόχος είναι η εύρεση της βέλτιστης διαδρομής, η οποία θα περνάει από όλες τις πόλεις και θα καταλήγει στην πόλη την οποία ξεκίνησε. Ωστόσο στο VRP δεδομένα είναι ένα σύνολο σημείων (συνήθως είναι διευθύνσεις πελατών) και μια αποθήκη(άλλο ένα σημείο το οποίο αποτελεί σημείο εκκίνησης) και στόχος είναι η εύρεση να ενός πλήθους βέλτιστων διάδρομων στις οποίες το όχημα ξεκινάει από το σημείο της αποθήκης που έχει οριστεί περνάει από όλα τα υπόλοιπα σημεία(διευθύνσεις πελατών) και επιστρέφει ξανά στην αποθήκη.

Το πρόβλημα της βέλτιστης δρομολόγησης οχήματος αφορά κυρίως εταιρείες που ασχολούνται με την υπηρεσία της αποστολής και παράδοσης προϊόντων. Ανάλογα την ειδική παραλλαγή μπορεί ο τελικός στόχος να είναι διαφορετικός, συνήθως όπως είναι στην κλασσική περίπτωση είναι η ελαχιστοποίηση του ολικού κόστους, άλλοτε όμως μπορεί να είναι η ελαχιστοποίηση των απαιτούμενων οχημάτων ή ελαχιστοποίηση της διακύμανσης του χρόνου και του όγκου μεταφοράς, η αποφυγή ποινών για υπηρεσίες χαμηλής ποιότητας ή η βελτιστοποίηση κέρδους ανάλογα των παραγόντων που δίνουν κέρδος.

Για την κλασσική μαθηματική διατύπωση του προβλήματος έχουμε:

- Έναν γράφο G $\{V,A,D\}$ όπου V $\{0,1,\ldots,n\}$ το σύνολο των κόμβων που αντιπροσωπεύουν τους πελάτες, $A\{(i,j):i,j\in Vi\neq j\}$ το σύνολο των ακμών που συνδέουν τους κόμβους και D $\{di,j:i,j\in Vi\neq j\}$ η απόσταση των κόμβων αυτών.
- Έναν στόλο που αποτελείται από N οχήματα, η χωρητικότητα του οχήματος είναι Q και το L_i αντιπροσωπεύει την ζήτηση στον εκάστοτε κόμβο i και με R συμβολίζεται η ακτίνα εμβέλειας του κάθε οχήματος.

Το γραμμικό μοντέλο αποτυπώνεται από το σύνολό των παρακάτω εξισώσεων:

$$min \sum_{k=1}^{N}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{i,j}^{k} \qquad (1)$$

$$\sum_{j=2}^{n} x_{1,j}^{k} = \sum_{j=2}^{n} x_{j,1}^{k} = 1\ \forall k = 1,\ldots N \qquad (2)$$

$$\sum_{k=1}^{N}\sum_{j=2}^{n} x_{i,j}^{k} = \sum_{k=1}^{N}\sum_{j=2}^{n} x_{j,i}^{k} = 1\ \forall i = 1,\ldots n \qquad (3)$$

$$\sum_{k=1}^{N}\sum_{i\in S}\sum_{j\notin S} x_{i,j}^{k} \geq 1 \forall S \subseteq \{1,2,\ldots n\} S \neq 0 \qquad (4)$$

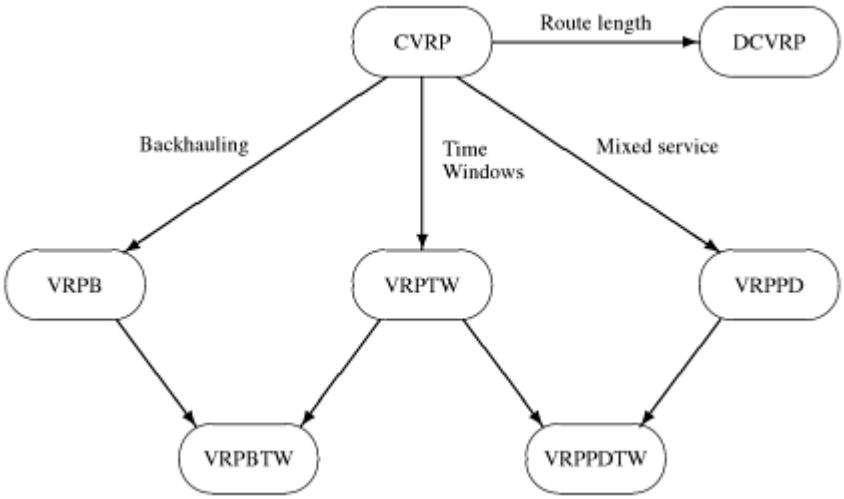
$$\sum_{i=1}^{n}\sum_{j=2}^{n} L_j\, x_{i,j}^{k} \leq Q \qquad (5)$$

$$\sum_{i=1}^{n}\sum_{j=2}^{n} d_{i,j}\, x_{i,j}^{k} \leq R \qquad (6)$$

Η Αντικειμενική συνάρτηση (1) εκφράζει της ελαχιστοποίηση του κόστους μεταφοράς του συνολικού στόλου. Ο Περιορισμός (2) εξασφαλίζει ότι κάθε όχημα – δρομολόγιο θα πρέπει να ξεκινήσει και να καταλήξει στην αποθήκη και να εξυπηρετήσει τουλάχιστον ένα πελάτη. Ο Περιορισμός (3) εκφράζει το γεγονός ότι όλοι οι πελάτες – κόμβοι εξυπηρετούνται ακριβώς μια φορά και ότι ένα μόνο όχημα θα φτάσει και θα φύγει από τον κόμβο αυτό. Η Συνθήκη (4) διασφαλίζει ότι δεν υπάρχουν βρόχοι στην λύση. Οι Περιορισμοί (5) και (6) εκφράζουν τον περιορισμό στην χωρητικότητα των οχημάτων καθώς και την χωρική εμβέλεια του δρομολογίου. (πηγή: βιβλιογραφία[5])

Παραλλαγές του κλασσικού προβλήματος VRP

Το VRP έχει πολλές άμεσες εφαρμογές στην βιομηχανία και η χρήση υπολογιστών για την βελτιστοποίηση διάφορων παραλλαγών ανάλογα των υπηρεσιών και των σκοπών της επιχείρησης μπορεί να αυξήσει σε ένα ποσοστό τα συνολικά κέρδη. Σε αυτό το κεφάλαιο λοιπόν θα δούμε μερικές από τις πιο σημαντικές παραλλαγές αυτού του προβλήματος.



Σχήμα 2.3: Βασικές παραλλαγές του VRP και οι μεταξύ τους διασυνδέσεις (Πηγή: Toth and Vigo, 2002)

Capacitated and Distance-Constrained VRP: Σε αυτή την εκδοχή τα οχήματα έχουν μια περιορισμένη χωρητικότητα σε αγαθά που μπορούν να μεταφέρουν, με περιορισμό την χωρητικότητα αυτή στόχος είναι να ελαχιστοποιηθεί το συνολικό κόστος εξυπηρέτησης πελατών. Το κόστος προκύπτει από το συνολικό μήκος και τον συνολικό χρόνο που απαιτούν οι διαδρομές που θα γίνουν για να εξυπηρετηθούν οι πελάτες. Αν ωστόσο υπάρχει περιορισμός και στην συνολική απόσταση που μπορεί να διανθεί τότε το πρόβλημα παίρνει την μορφή του Distance - Constraind VRP (DCVRP).
(πηγή: βιβλιογραφία[5][6])

VRP with Time Windows: Σε αυτή την παραλλαγή κάθε σημείο παράδοσης έχει και ένα χρονικό περιθώριο στο οποίο θα πρέπει να γίνει η παράδοση. Οι πελάτες λοιπόν θα πρέπει να

εξυπηρετούνται εντός αυτού του χρονοπαραθύρου και το όχημα να σταματάει για συγκεκριμένο χρόνο κάθε φορά(χρόνος εξυπηρέτησης). (πηγή: βιβλιογραφία[5][7])

VRP with Backhauls: Σε αυτή την παραλλαγή χωρίζεται το σύνολο των πελατών σε δυο σύνολα, το πρώτο σύνολο ,έστω L, αποτελείται από τους πελάτες εμπρόσθιας εξυπηρέτησης (Linehaul customers) στους οποίους πρέπει να παραδοθούν αγαθά και το δεύτερο ,έστω B, από τους πελάτες οπίσθιας εξυπηρέτησης (Backhaul customers) από τους οποίους πρέπει να παραληφθούν αγαθά. Δίνεται προτεραιότητα στην παράδοση και όχι στην παραλαβή δηλαδή προτεραιότητα έχουν οι πελάτες του συνόλου L. (πηγή: βιβλιογραφία[5])

VRP with Pick Up and Delivery: Σε αυτή την περίπτωση κάθε πελάτης συνδέεται με μια παραλαβή μιας συγκεκριμένης ποσότητας αγαθών αλλά και με μια επιστροφή και για κάθε πελάτη υπάρχει μια αποθήκη παραλαβής και μια αποθήκη αποστολής. Για κάθε πελάτη χρονικά πρώτα γίνεται η παράδοση φορτιού και έπειτα η επιστροφή φορτιού από τον πελάτη. (πηγή: βιβλιογραφία[5])

Μερικές ακόμα παραλλαγές που στην βιβλιογραφία τους συναντάμε ενδιαφέροντα στοιχεία που μας βοηθάνε και στο δικό μας πρόβλημα είναι:

- Open Vehicle Routing Problem (OVRP): Σε αυτή την παραλλαγή τα οχήματα δεν είναι αναγκαίο να επιστρέψουν στην κεντρική αποθήκη στο τέλος της διαδρομής. (πηγή: βιβλιογραφία[8])
- Multi-Depot Vehicle Routing Problem (MDVRP): Σε αυτή την παραλλαγή έχουμε πολλαπλές αποθήκες από τις οποίες τα οχήματα μπορούν να ξεκινήσουν και να τελειώσουν τις διαδρομές τους. (πηγή: βιβλιογραφία[9])

2.3 Ευρετικοί αλγόριθμοι και τεχνικές προσπάθειας εύρεσης βέλτιστης λύσης δρομολόγησης

Για την επίλυση των TSP και VRP και των παραλλαγών τους που είδαμε πιο πάνω ανά τα χρονιά έχουν εφευρεθεί διάφορες τεχνικές επίλυσης τους, εδώ θα κάνουμε μια ανασκόπηση στους πιο σημαντικούς αλγορίθμους.

Αρχικά έχουμε την κατηγορία των ευρετικών και μεθευρετικών αλγορίθμων με μερικούς από τους πιο σημαντικούς να είναι οι εξής:

- Simulated Annealing (SA)
- Πλησιέστερος Γείτονας (Nearest Neighbor - NN)
- Άπληστος Αλγόριθμος (Greedy Heuristic Algorithm – GH)
- Αναζήτηση Ταμπού (Tabu Search – TS)
- Βελτιστοποίηση Αποικίας Μυρμηγκιών (Ant Colony Optimization – ACO)
- Γενετικό Αλγόριθμο (Genetic Algorithm – GA)
- Βελτιστοποίηση Σμήνους Σωματιδίων (Particle Swarm Optimization – PSO)
  (πηγή: βιβλιογραφία[10][11][12])

Όταν οι αλγόριθμοι αυτοί συνδυάζονται με αλγορίθμους τοπικής αναζήτησης για να πάρουμε ακόμα πιο αποδοτικές μεθόδους βελτιστοποίησης διαδρομής όπως είναι και ο αλγόριθμος Lin-Kernighan. (πηγή: βιβλιογραφία[13])

Οι ευρετικοι αλγόριθμοι λειτουργούν στα πλαίσια του προβλήματος που ορίζονται, βρίσκουν στοιχεία, ιδιότητες και ιδιαιτερότητες γύρω από το πρόβλημα και προσπαθούν να έρθουν σε μια καλή λύση. Το πρόβλημα τους είναι ότι είναι υπερβολικά άπληστη, δηλαδή ψάχνουν συνεχώς βελτιώσεις, με αποτέλεσμα να παγιδεύονται συχνά σε τοπικά μέγιστα.

Για την αντιμετώπιση αυτού του προβλήματος έχουμε την ύπαρξη των μεθευρετίκων αλγορίθμων οι οποίοι είναι ανεξάρτητοι από το πρόβλημα που ορίζεται, δηλαδή δεν χρησιμοποιούν ιδιαιτερότητες του. Οι μεθευρετικοί αλγόριθμοι μπορούν προσωρινά να δεχτούν και μια λύση που είναι χειρότερη προκειμένου να προσπαθήσουν να ψάξουν σε όλο το εύρος των πιθανών λύσεων, την βέλτιστη λύση. Με αυτό τον τρόπο είναι πιο δύσκολο να παγιδευτούν σε τοπικά μέγιστα. (πηγή: βιβλιογραφία[14])

Μια σύντομη παρουσίαση των ευρετικών αλγορίθμων του πλησιέστερου γείτονα και του γενετίκου αλγοριθμού που συναντώνται συχνά:

Πλησιέστερος Γείτονας

Γνωστός και ως Nearest Neighbor (NN) στην διεθνή βιβλιογραφία. Είναι ο πιο απλός αλγόριθμος ο οποίος θα μπορούσε κάλλιστα να είναι και η διαισθητικά πρώτη στρατηγική αντιμετώπισης κάποιου που θα βρισκόταν στα πλαίσια ενός τέτοιου προβλήματος (TSP).
Τα βήματα είναι τα εξής:

1. Διαλέγουμε τυχαία μια πόλη n και την ονομάζουμε n_0
2. Πάμε στην πλησιέστερη πόλη που δεν έχουμε επισκεφθεί.
3. Διαγράφουμε αυτή την πόλη από το σύνολο των πόλεων που πρέπει να επισκεφθούμε
4. Εάν υπάρχουν ακόμα πόλεις που δεν έχουμε επισκεφθεί πάμε στο βήμα 2
5. Επιστρέφουμε στην αρχική μας πόλη (πόλη n_0) (πηγή: βιβλιογραφία[15])

Γενετικός Αλγόριθμος

Ο Γενετικός Αλγόριθμος (Genetic Algorithm – GA) στην επιστήμη των υπολογιστών αλλά και την τεχνητή νοημοσύνη συναντάται συχνά για την προσπάθεια εύρεσης λύσεως σε δύσκολα προβλήματα . Η έμπνευση για την δημιουργία του είναι από την επιστήμη της βιολογίας, για αυτό και η ονομασία αλλά και πολλές ορολογίες που συναντάμε στην λειτουργία του έχουν ονομάσιες από αυτή την επιστήμη. Για την κύρια φιλοσοφία της λειτουργίας (στην βιβλιογραφία συναντάμε πολλές παραλλαγές και προσθήκες) του μπορούμε να πούμε ότι:

- Ο αλγόριθμος υπολογίζει μια συνάρτηση καταλληλότητας για κάθε κόμβο του αρχικού δικτύου.
- Στη συνέχεια, δημιουργεί νέους κόμβους. Χρησιμοποιεί τη διαδικασία της μετάλλαξης για να προσθέσει τυχαιότητα στη διαδικασία, παρόμοια με αυτή του φυσικού γονιδιώματος.
- Τέλος, επιλέγει τη λύση εκείνη με την υψηλότερη συνάρτηση καταλληλότητας.

Η εφαρμογή του Γενετικού Αλγορίθμου στο TSP εμφανίζει μερικές ιδιαιτερότητες. Για παράδειγμα, σε κάθε διαδρομή, κάθε πόλη δεν πρέπει να επαναλαμβάνεται, διαφορετικά εμφανίζονται βρόχοι (loops). Επιπλέον, κατά την εκτέλεση του αλγορίθμου εξετάζονται μόνο έγκυρες διαδρομές.

Οι Naveen Kumar, Karambir & Rajiv Kumar έκαναν μια ερευνά στην χρήση γενετικών αλγορίθμων για την επίλυση του TSP. Αφού πραγματοποίησαν αναλύσεις χρησιμοποιώντας μια γκάμα τεχνικών που βρίσκουμε στους γενετικούς αλγορίθμους(crossover,mutation και συνδυασμος των δυο) για την εύρεση λύσης στο TSP παρατήρησαν ότι θα μπορούσαν να βρεθούν και άλλες γενετικές τεχνικές για βελτίωση της απόδοσης. (πηγή: βιβλιογραφία[16])

2.3.1 Τοπική αναζήτηση και αλγόριθμοι ανταλλαγής ζευγών

Τοπική αναζήτηση

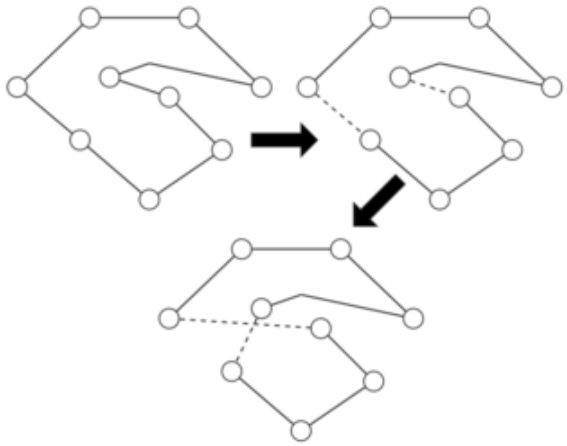
Η διαδικασία της τοπικής αναζήτησης ανήκει και αυτή στην κατηγορία των ευρετικών αλγορίθμων, η ιδέα της οποίας έχει προκύψει από την ανθρώπινη λογική. Μια τέτοια διαδικασία εφαρμόζεται σε μια ήδη υπάρχουσα λύση και εκτελείται μια σειρά διαδοχικών εφικτών αλλαγών έτσι ώστε να βρεθεί μια ακόμα καλύτερη λύση. Τέτοιες μέθοδοι δεν εγγυούνται ότι θα μπορέσουν να βρουν κάποια βελτίωση της ήδη υπάρχουσας λύσης, για το λόγο αυτό η στρατηγική με την οποία χρησιμοποιούνται είναι καθοριστικής σημασίας στα αποτελέσματα που θα δώσουν. Ο συνδυασμός διαφορετικών τεχνικών τοπικής αναζήτησης συχνά δίνει καλύτερη πιθανότητα να βρεθεί καλύτερη λύση. (πηγή: βιβλιογραφία[17]).

Για την σχεδίαση ενός αλγορίθμου τοπικής αναζήτησης λοιπόν πρέπει να λάβουμε υπόψιν:

- Την επιλογή της γειτονίας αναζήτησης, δηλαδή ενός συνόλου σημείων που βρίσκονται εντός μιας απόστασης d οπού έχουν επιλεχθεί από την δομή του προβλήματος, στα οποία θα εφαρμόσουμε μια τεχνική τοπικής αναζήτησης.
- Έπειτα την ποιότητα της αρχικής λύσης. Πριν αρχίσουμε να ψάχνουμε για βελτιώσεις είναι σημαντικό η αρχική μας λύση να είναι όσο το δυνατόν καλύτερη γιατί αυξάνονται οι πιθανότητες να βρούμε ευκολότερα και γρηγορότερα ακόμα καλύτερη λύση.
- Και το πιο σημαντικό είναι η μέθοδος που χρησιμοποιείται για την βελτίωση της αρχικής λύσης. Πολλές από τις μεθοδολογίες εμφανίζουν προβλήματα εύρεσης καλής λύσης στο σύνολο των πιθανών λύσεων διότι παγιδεύονται σε τοπικά ελάχιστα. Μια από τις πιο αποδοτικές μεθόδους που αποφεύγουν να κολλήσουν σε τοπικά μέγιστα είναι η μέθοδος Lin – Kernighan την οποία και θα χρησιμοποιήσουμε για την λύση του προβλήματος μας. (πηγή: βιβλιογραφία[18])

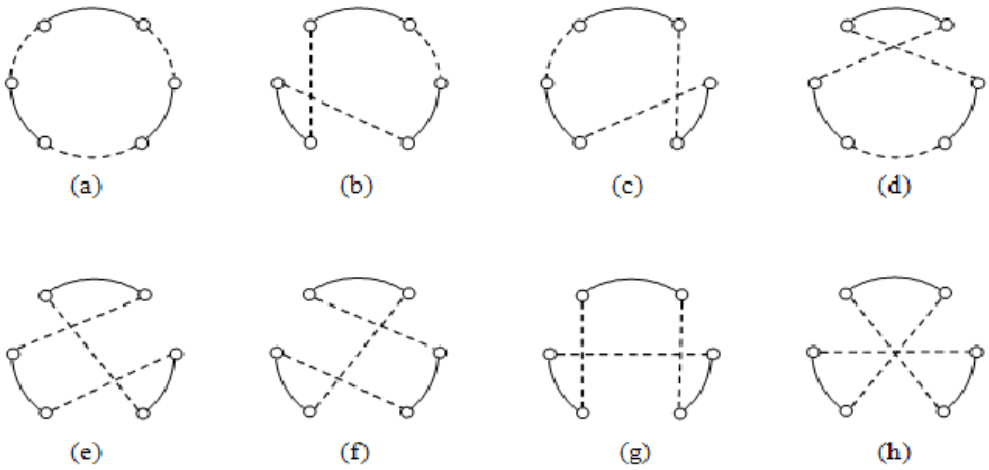
Αλγόριθμοι ανταλλαγής ζευγών(Pairwise Exchange – 2opt)

Αποτελούν αλγορίθμους που διαλέγουν δυο πόλεις και τις διαδρομές τους, και ανταλλάσσουν τις διαδρομές για την δημιουργία μιας νέας τελικής διαδρομής. Σκοπός είναι να βρεθούν ζεύγη πόλεων που άμα γίνει αυτή η ανταλλαγή η τελική διαδρομή να είναι μικρότερη.



Σχημα 2.4 Παράδειγμα επανάληψης αλγορίθμου 2-opt

Είναι ένας αλγόριθμος που λειτουργεί με διαδοχικές επαναλήψεις. Με την ιδιά τεχνική λειτουργεί και ο αλγόριθμος 3-opt οπού διαλέγει 3 πόλεις και τις διαδρομές τους και δοκιμάζει καινούργιους συνδυασμούς διάδρομων από ανταλλαγή διάδρομων.



Σχημα 2.5 Όλες οι πιθανές 3-opt ανταλλαγές ενός TSP με έξι πόλεις

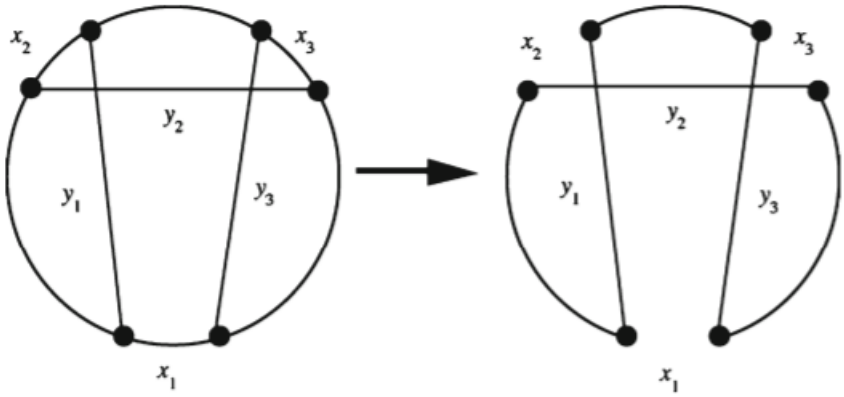
Αυτές είναι πιο συνηθισμένες τεχνικές που χρησιμοποιούνται για την εύρεση μιας αρκετά καλής λύσης.

Επίσης έχουμε και τις τεχνικές k-opt (Lin-Kernighan heuristic), την τεχνική v-opt heuristic, τεχνικές που ψάχνουν τυχαία βελτιώσεις και τα τελευταία χρονιά έχουν εμφανιστεί και τεχνικές τεχνητής νοημοσύνης για την εύρεση κάλων λύσεων. (πηγή: βιβλιογραφία[19][20])

2.3.2 Αλγόριθμος των Lin- Kernighan

Ο αλγόριθμος Lin – Kernighan ανήκει σε αυτό που λεμέ αλγόριθμοι τοπικής αναζήτησης (local search algorithms). Ένας αλγόριθμος τοπικής αναζήτησης ξεκινάει από μια κατάσταση στο πλαίσιο της αναζήτησης και με διαδοχικές κινήσεις πάει σε μια

γειτονική κατάσταση.



Σχημα 2.6 Μια κίνηση 3 – opt

Ο αλγόριθμος αποτελείται από κινήσεις που μπορούν να μετατρέψουν την ήδη υπάρχουσα λύση σε μια άλλη. Δοσμένου ενός προβλήματος TSP , ο αλγόριθμος επαναλαμβανόμενα εκτελεί ανταλλαγές με στόχο την μείωση της συνολικής διαδρομής, μέχρι να μην μπορεί να βρει άλλη βελτίωση. Η διαδικασία μπορεί να επαναληφθεί πολλές φορές από αρχικές διαδρομές που έχουν δημιουργηθεί από μια τυχαία επιλογή.

Ο αλγόριθμος Lin – Kernighan εκτελεί αυτό που λεμέ k – opt κινήσεις σε μια διαδρομή. Μια k – opt ανταλλαγή είναι μια η οποία αντικαθιστά k κόμβους (edges) της διαδρομής με k άλλους κόμβους με στόχο να βρεθεί μια συντομότερη διαδρομή.

Αν T είναι η τωρινή διαδρομή, σε κάθε επανάληψη ο αλγόριθμος προσπαθεί να βρει δυο ομάδες κόμβων, , X = {x_1,..., x_k } και Y = {y_1,..., y_k } τέτοιες ώστε αν οι κομβόι X διαγράφουν από την διαδρομή T και αντικατασταθούν από τους κόμβους Y το αποτέλεσμα θα είναι μια πιο σύντομη διαδρομή.

Τα δυο σετ κόμβων X και Y δημιουργούνται προστίθοντας έναν κόμβο την φορά. Αρχικά είναι άδεια. Στο βήμα i ένα ζευγάρι κόμβων, x_i και y_i, προστίθενται στο σύνολο X και Y.

Προκειμένου να επιτευχθεί ένας ιδιαίτερα αποδοτικός αλγόριθμος, μόνο κόμβοι που πληρούν τα παρακάτω κριτήρια μπορούν να εισέλθουν στα σύνολα X και Y:

1) Το κριτήριο διαδοχικής ανταλλαγής: τα x_i και y_i πρέπει να μοιράζονται ένα τελικό σημείο, και το ίδιο πρέπει να ισχύει και για τα y_i και x_{i+1}. Αν ως t_1 δηλώνεται ένα από

τα δύο τελικά σημεία του x_1, έχουμε γενικά: $x_i = (t_{2i-1}, t_{2i})$, $y_i = (t_{2i}, t_{2i+1})$ και $x_{i+1} = (t_{2i+1}, t_{2i+2})$ για $i \geq 1$.

2) Το κριτήριο σκοπιμότητας: Απαιτείται να επιλέξουμε το $x_i = (t_{2i-1}, t_{2i})$ έτσι ώστε, αν το t_{2i} είναι ενωμένο με το t_1, τότε η ένωση που θα προκύψει να είναι μια διαδρομή. Αυτό το κριτήριο σκοπιμότητας χρησιμοποιείται για $i \geq 3$ και εγγυάται ότι είναι δυνατή η ολοκλήρωση μιας διαδρομής. Αυτό το κριτήριο συμπεριλήφθηκε στον αλγόριθμο τόσο για τη μείωση του χρόνου λειτουργίας όσο και για την απλοποίηση της κωδικοποίησης.
3) Το κριτήριο θετικού κέρδους: Απαιτείται ότι το yi επιλέγεται πάντα έτσι ώστε το κέρδος Gi από το προτεινόμενο σύνολο ανταλλαγών να είναι θετικό. Έστω ότι το gi = c(xi) – c(yi) είναι το κέρδος από την ανταλλαγή του xi με το yi. Τότε Gi είναι το άθροισμα g1 + g2 + .. + gi. Το κριτήριο αυτό παίζει σημαντικό ρόλο στην αποτελεσματικότητα του αλγορίθμου. Η απαίτηση ότι κάθε μερικό άθροισμα, Gi, πρέπει να είναι θετικό, φαίνεται αρχικά να είναι πολύ περιοριστική όμως, δεν ισχύει όπως φαίνεται και από το ακόλουθο παράδειγμα: Εάν μια ακολουθία αριθμών έχει ένα θετικό άθροισμα, υπάρχει μια κυκλική μεταβολή αυτών των αριθμών έτσι ώστε κάθε μερικό άθροισμα να είναι θετικό. Η απόδειξη είναι απλή και μπορεί να βρεθεί στην εργασία των Lin & Kernighan
4) Το κριτήριο της διαζευκτικότητας: Τέλος, είναι και πάλι επιθυμητό τα σύνολα X και Y να είναι διακριτά. Αυτό απλοποιεί την κωδικοποίηση, μειώνει το χρόνο επίλυσης και δίνει στον αλγόριθμο ένα κριτήριο πραγματικής διακοπής. (πηγή: βιβλιογραφία[21],[13])

2.4 Η ομαδοποίηση K μέσων (K – means clustering)

Στο πρόβλημα μας εκτός από την εύρεση της βέλτιστης διαδρομής που θα πρέπει να ακολουθήσει ο επιθεωρητής για την επίσκεψη ενός πεπερασμένου αριθμού στάσεων, τίθεται και το ζήτημα πως θα υποδιαιρεθεί το σύνολο των στάσεων του δικτύου ΟΑΣΑ. Αυτή η διαδικασία είναι απαραίτητη καθώς δεν είναι δυνατή η επίσκεψη του σύνολο των στάσεων σε μια εργάσιμη ημερά. Επίσης έχει ιδιαίτερη σημασία η όσο το δυνατόν καλύτερη

υποδιαίρεση του συνόλου έτσι ώστε να είναι εφικτό και οι στάσεις σε κάθε υποσύνολο να μην απέχουν μεγάλες αποστάσεις. Για το στόχο μας αυτό θα χρησιμοποιήσουμε την τεχνική της ομαδοποίησης ή της συσταδοποίησης με k-means.

Ο αλγόριθμος k-means χρησιμοποιήθηκε για πρώτη φορά από τον James MacQueen το 1967 και από τότε έχει χρησιμοποιηθεί αρκετά σε διαδικασίες ψηφιακής επεξεργασίας σήματος. Η ομαδοποίηση μέσω αυτού του αλγορίθμου έχει ως στόχο να διαχωρίσει n παρατηρήσεις σε K ομάδες, έτσι ώστε κάθε παρατήρηση να ανήκει στη συστάδα με το κοντινότερο μέσο, το οποίο χρησιμεύει ως ένα χαρακτηριστικό δείγμα της συστάδας. Αυτό οδηγεί σε μια διαμερίσει του χώρου σε κελιά Voronoi.

Τα δεδομένα που δεχόμαστε στον αλγόριθμο είναι ένα σύνολο προτύπων x_1 ,x_2 , …,x_n (στην δικιά μας περίπτωση οι στάσεις) και ο επιθυμητός αριθμός των ομάδων K.

Ξεκινάμε λοιπόν μια επαναληπτική διαδικασία οπού:

- Αναθέτουμε τυχαία τα κέντρα των ομάδων
- Για κάθε στοιχείο X_i υπολογίζουμε το κοντινότερο κέντρο K_j και το αναθέτουμε στην συστάδα του.
- Στη συνέχεια για κάθε συστάδα K_1,K_2,…,K_n υπολογίζουμε τα νέα γεωμετρικά τους κέντρα ως τους μέσους ορούς κάθε συστάδας με όλα τα σημεία x_i που προέκυψαν από το προηγούμενο βήμα.
- Σταματάμε την διαδικασία όταν δεν έχουμε μεταβολές στις συστάδες

Ο αλγόριθμος k-means είναι πολυπλοκότητας O(l*n*K*d) με l τον αριθμό των επαναλήψεων, K τον αριθμό των συστάδων, n τον αριθμό των σημείων, d τον αριθμό των διαστάσεων.

Ο αλγόριθμος αυτός είναι κατάλληλος για την χρήση στο πρόβλημα μας διότι εξασφαλίζει ότι στην ομαδοποίηση που θα συμβεί οι στάσεις μας θα απέχουν όσο το δυνατόν λιγότερο γίνεται οπότε η επιθεώρηση τους θα είναι πιο εύκολη καθώς το σύνολο των χιλιομέτρων που θα διανθεί θα είναι μικρότερο.

Η μαθηματική διατύπωση του αλγορίθμου έχει ως εξής:

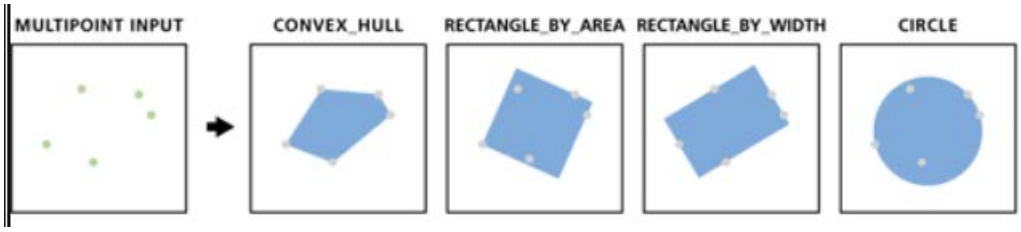
-Αναθέτουμε τυχαία τα κέντρα των ομάδων K_1 , K_2 ,…,Kn

- ∀ x_i υπολογίζουμε το κοντινότερο κέντρο Kj : $argminjD\ xi\ ,Kj$ και αναθέτουμε το x_i στην συστάδα K_j

- Στη συνέχεια για κάθε συστάδα K_1 , K_2 ,…,K_n υπολογίζομε τα νέα γεωμετρικά τους κέντρα: K_j = 1 n_j x_i→K_j (πηγή: βιβλιογραφία[22][23])

2.5 Η έννοια της ελάχιστης περιβάλλουσας γεωμετρίας (minimum bounding geometry) και η έννοια της χωρικής πυκνότητας σημείων

Στην γεωμετρία η έννοια της ελάχιστης περιβάλλουσας γεωμετρίας (minimum bounding geometry or minimum bounding box) για ένα σύνολο σημείων M σε ένα χώρο N διαστάσεων είναι η διαδικασία κατά την οποία βρίσκεται το ελάχιστο γεωμετρικό σχήμα, βάσει εμβαδού για δυο διαστάσεις, το οποίο περιλαμβάνει όλα τα σημεία. Θα κάνουμε χρήση αυτής της ιδιότητας στην συνέχεια για να αποκτήσουμε μια αντίληψη πόσο πυκνά είναι τοποθετημένες οι στάσεις στον χάρτη μας. (πηγή: βιβλιογραφία[23])

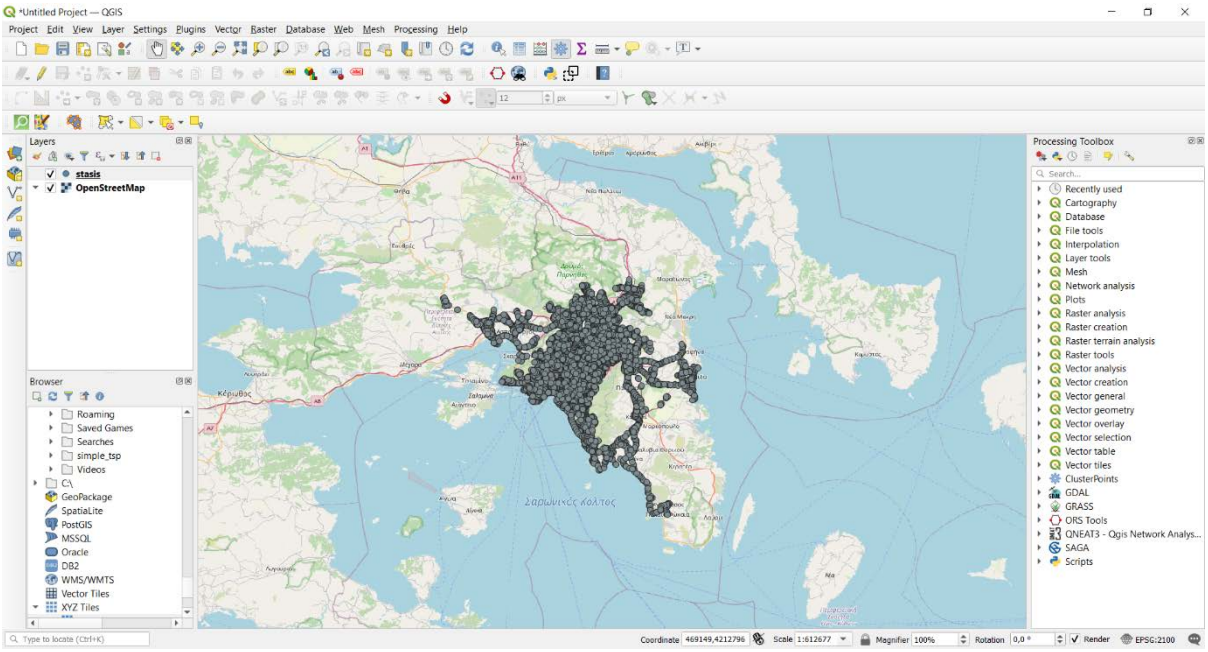


Σχήμα 2.7 Ελάχιστες περιβάλλουσες γεωμετρίες για ένα σύνολο σημείων

3. Λογισμικό Εφαρμογής και Λειτουργίες του

3.1 QGIS

Το Quantum GIS ή QGIS (http://www.qgis.org) είναι ένα ΕΛ/ΛΑΚ Σύστημα Γεωγραφικών Πληροφοριών.



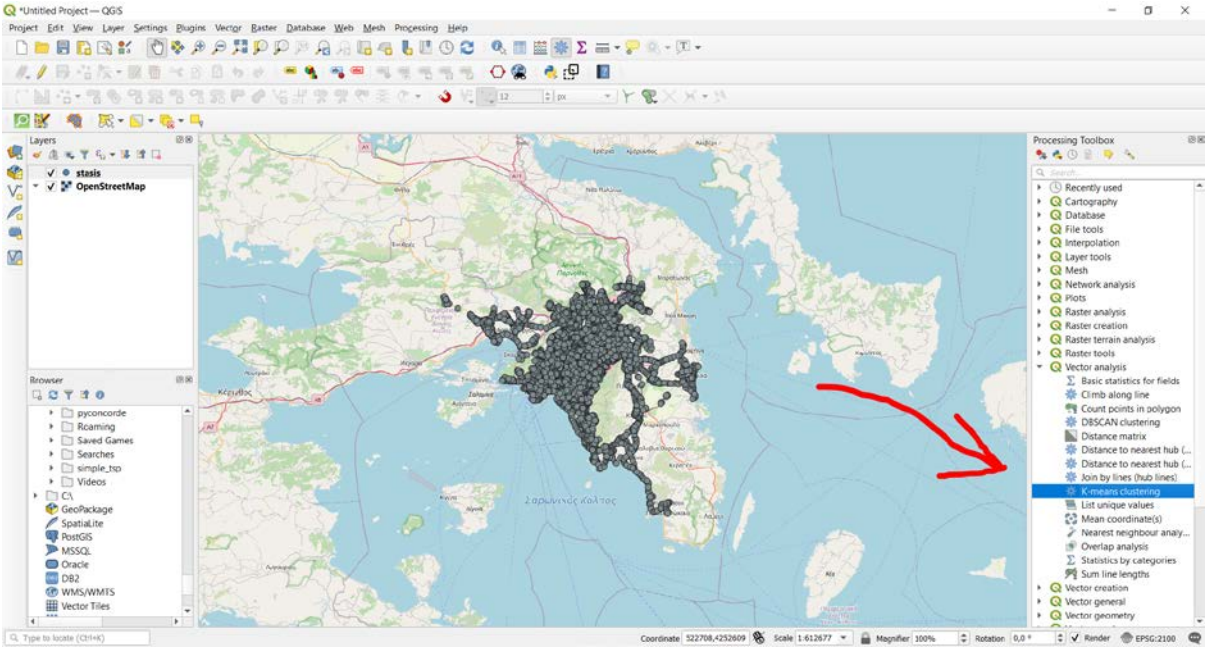
Σχημα 3.1 Το γραφικό περιβάλλον του QGIS

Το QGIS διαθέτει γραφικό περιβάλλον επικοινωνίας με το χρήστη και ενσωματώνει εργαλεία και λειτουργίες όπως η σχεδίαση και η ταυτόχρονη σχεδίαση διανυσματικών και κανονικοποιημένων γεωχωρικών δεδομένων, η διαχείριση και ο μετασχηματισμός του Συστήματος Αναφοράς Συντεταγμένων (ΣΑΣ), η διερεύνηση των δεδομένων και η χαρτοσύνθεση, η συλλογή, η επεξεργασία, η διαχείριση και η εξαγωγή, η χωρική ανάλυση και η γεωεπεξεργασία των δεδομένων, η δημοσιοποίηση στο διαδίκτυο κ.ά. Υποστηρίζει πολλαπλούς μορφότυπους διανυσματικών και κανονικοποιημένων δεδομένων και επικοινωνεί με χωρικές βάσεις δεδομένων.

3.1.1 K – means Clustering

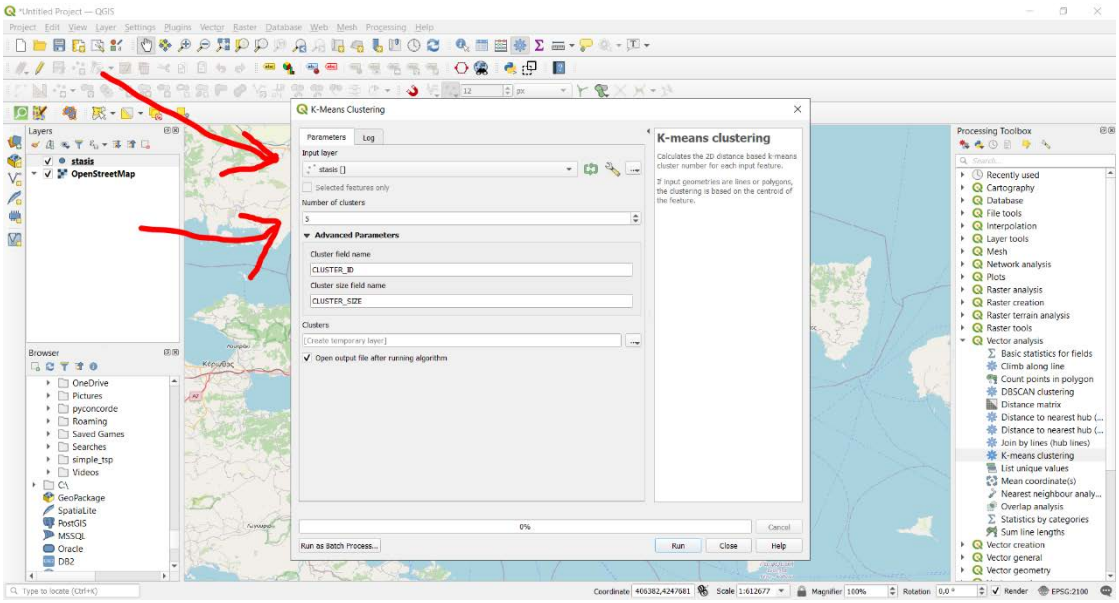
Τον αλγόριθμο του K – means θα τον χρησιμοποιήσουμε εκτενώς στην εργασία αυτή παρακάτω φαίνεται η αλληλουχία των βημάτων:

Αρχικά πάμε διαδοχικά Processing toolbox → vector analysis → k – means clustering



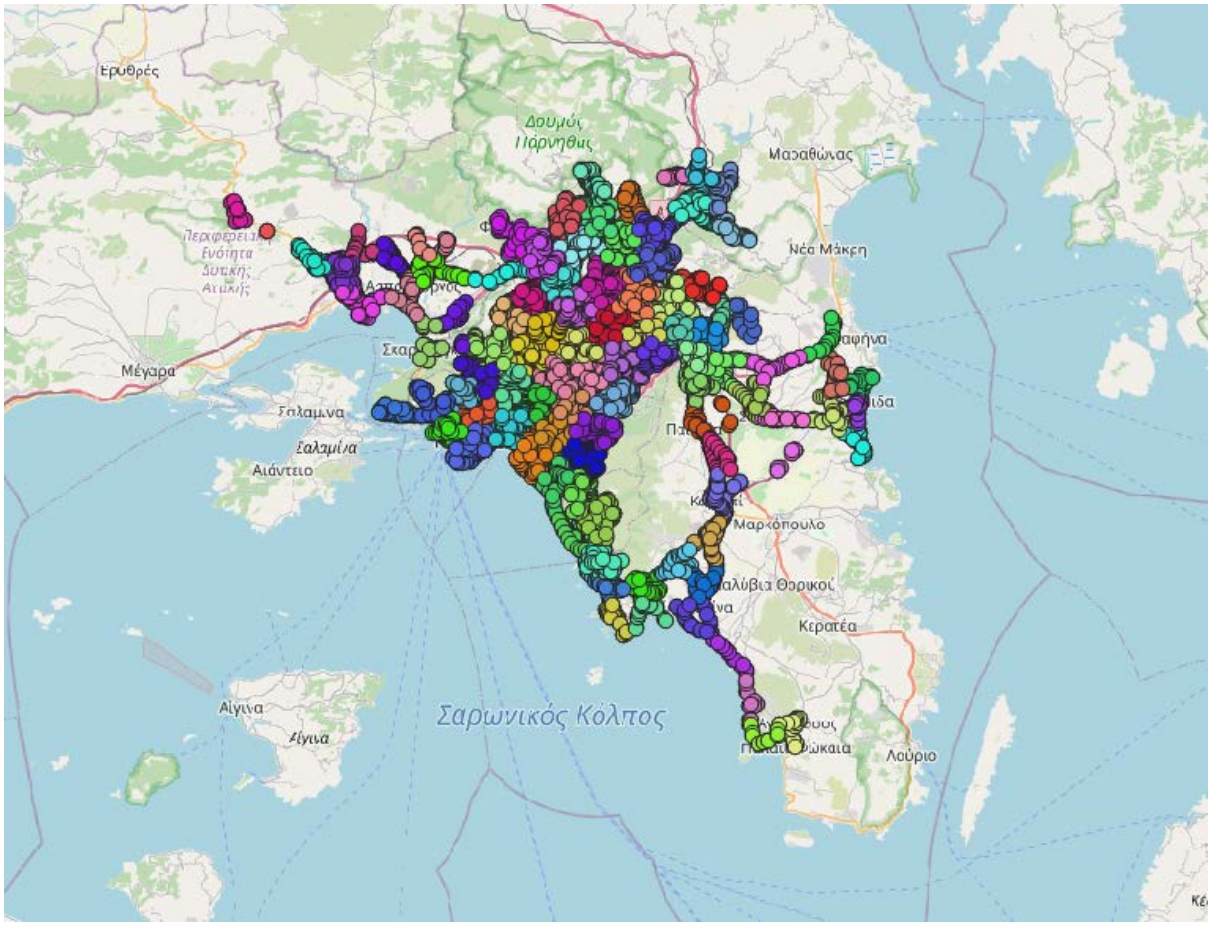
<u>Σχημα 3.2</u>

Και αφού ανοίξουμε το μενού του αλγορίθμου βλέπουμε την εξής εικόνα:



<u>Σχήμα 3.3</u>

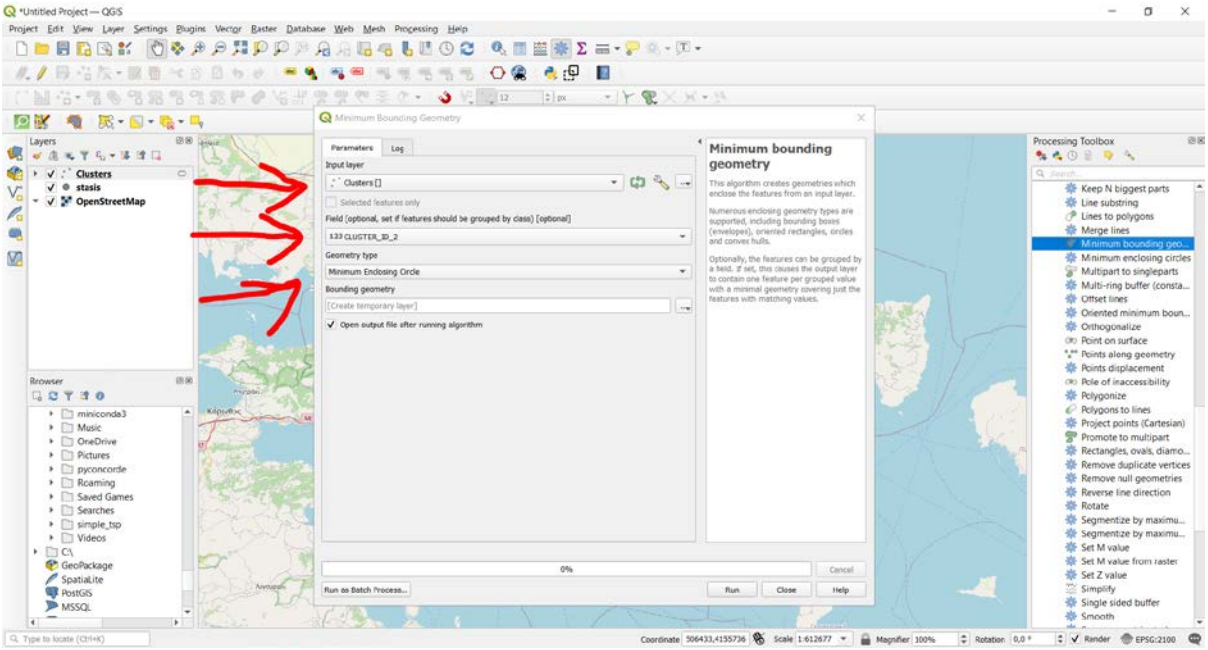
Όπου στο input layer συμπληρώνουμε το σύνολο των σημείων στην περίπτωση μας που θέλουμε να ομαδοποιήσουμε σε συστάδες και στο Number of clusters τον αριθμό των τελικών συστάδων που θα θέλαμε να δημιουργήσουμε από το εισαγόμενο σύνολο σημείων.



Σχήμα 3.4 Χάρτης που προέκυψε από 100 συστάδες του αλγορίθμου k-means

3.1.2 Minimum Bounding Geometry

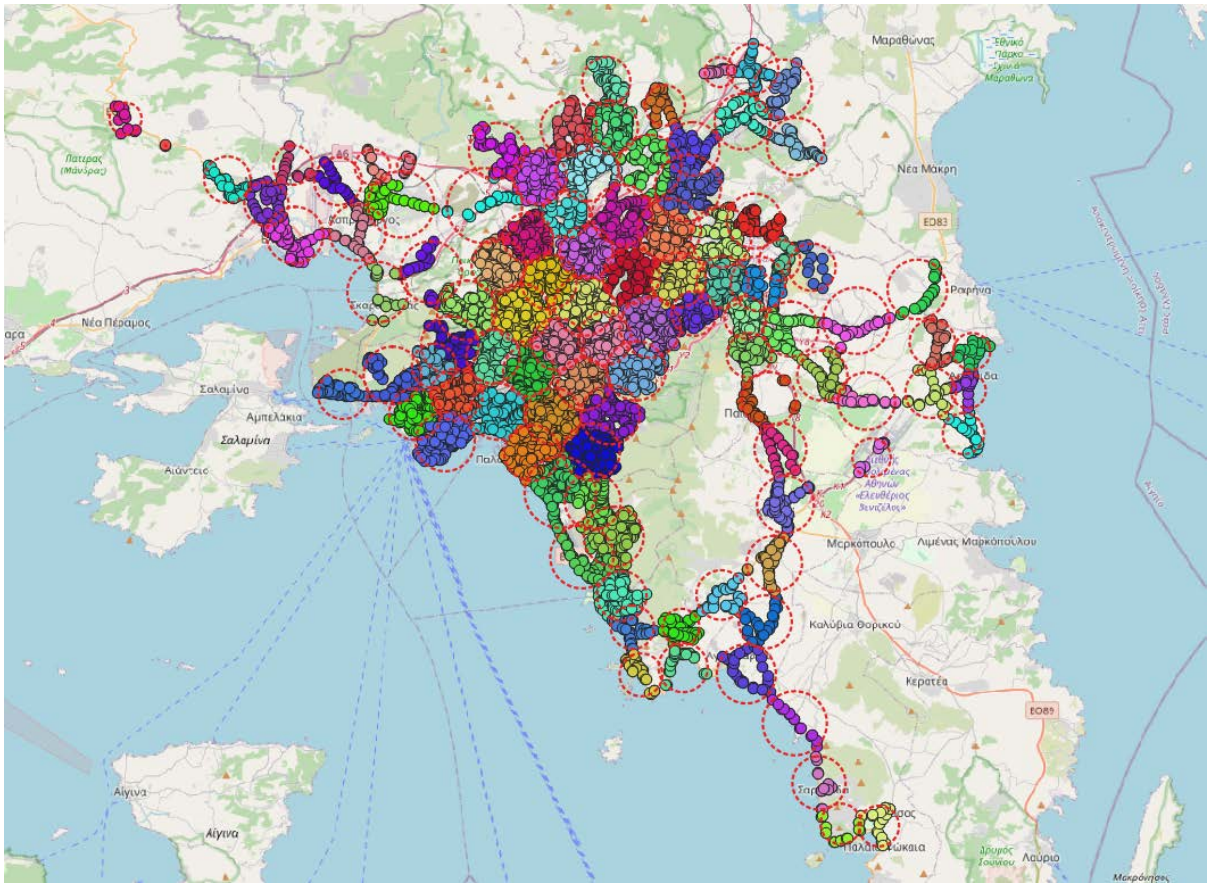
Ένα ακόμα εργαλείο που θα χρησιμοποιήσουμε είναι αυτό που λέγεται Minimum Bounding Geometry. Αφού περιηγηθούμε Processing toolbox → vector geometry →minimum bounding geometry και ανοίξουμε το μενού βλέπουμε τα εξής:



Σχήμα 3.5

- Στην κατηγορία input layer συμπληρώνουμε τα στοιχεία για τα οποία θέλουμε να βρούμε ελάχιστη περιβάλλουσα γεωμετρία.
- Στην κατηγορία Field (η οποία είναι προαιρετική) διαλέγουμε αν θέλουμε να ομαδοποιήσουμε αυτά τα στοιχεία βάσει κάποιου χαρακτηριστικού τους και να τρέξουμε τον αλγόριθμο επαναληπτικά για κάθε ομάδα αυτού του χαρακτηριστικού. Εμείς εδώ δηλαδή για τις 100 συστάδες που δημιουργήσαμε προηγουμένως θα βάλουμε στην κατηγορία field το Cluster_ID και ο αλγόριθμος μας θα βρει ελάχιστες γεωμετρίες για κάθε μια από τις 100 συστάδες του συνολικού μας δικτύου.
- Στην κατηγορία Geometry type διαλέγουμε αναμεσά από τις επιλογές bounding box (envelope), oriented rectangle, circle, convex hull την γεωμετρία που θέλουμε. Εμείς θα χρησιμοποιήσουμε αυτή του ελάχιστου περιβάλλοντα κύκλου.

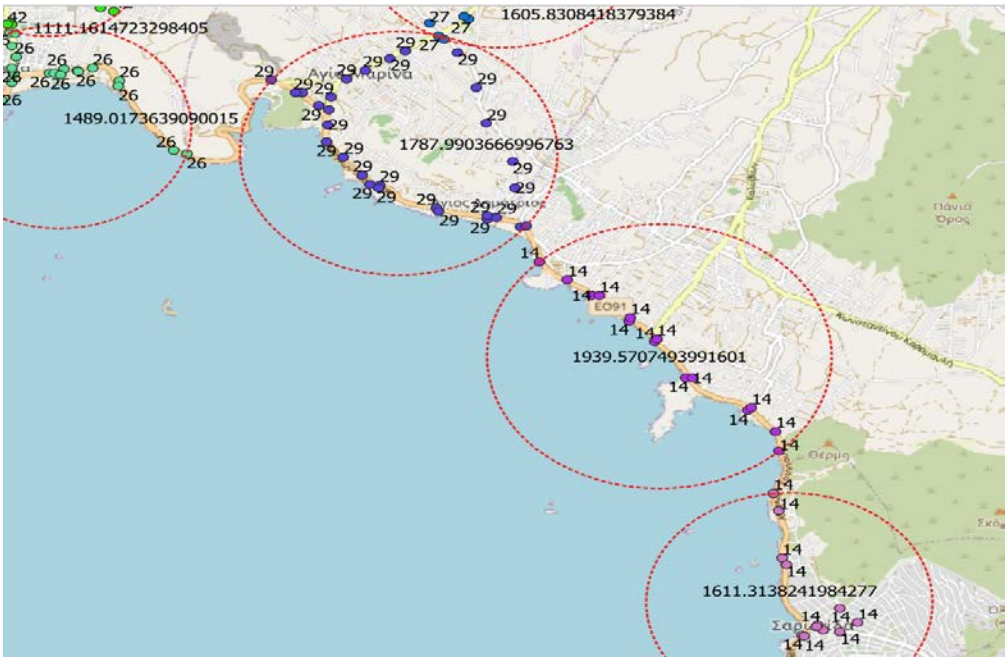
Οπότε τελικά παίρνουμε τον χάρτη:



Σχήμα 3.6 Χάρτης με τον ελάχιστα περιβάλλοντα κύκλο των 100 συστάδων

Όπου βλέπουμε ότι σε κάθε συστάδα έχουμε βρει τον κύκλο που την περικλείει (κόκκινη διακεκομμένη γραμμή).

Το πλεονέκτημα αυτό είναι ότι τώρα αν μεγεθύνουμε σε ένα μέρος του χάρτη βλέπουμε:



Σχήμα 3.7

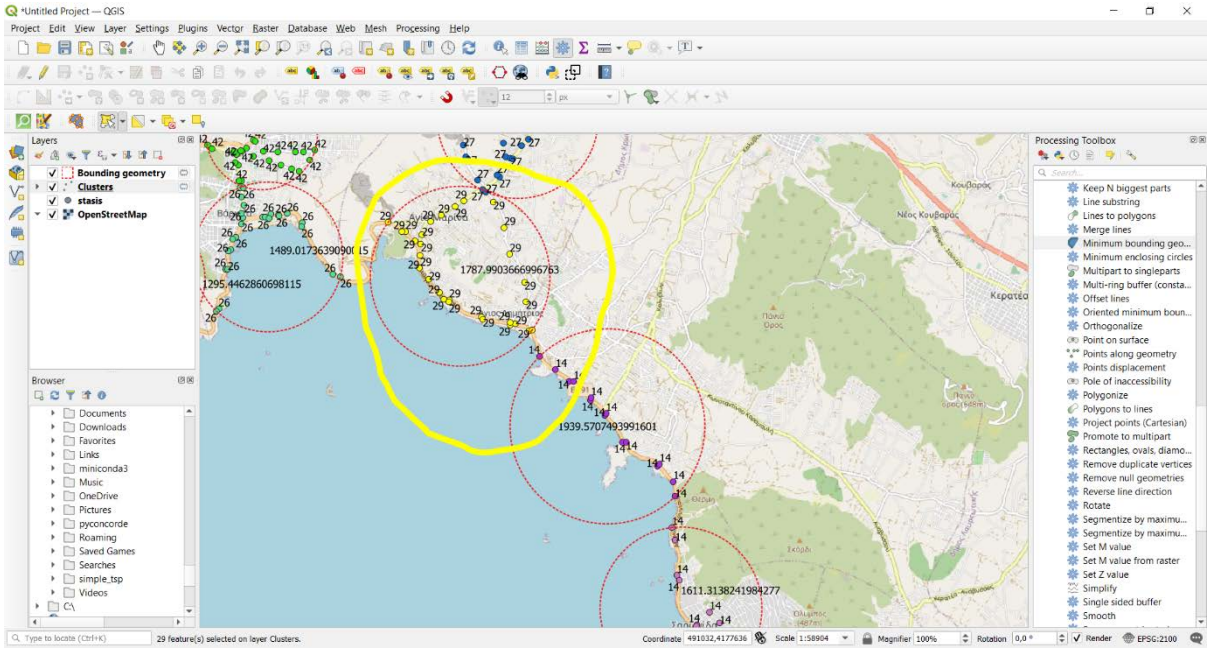
Όπου μπορούμε να διακρίνουμε την ακτίνα του κύκλου που περικλείει την συστάδα καθώς και τον αριθμό των στοιχείων που περιέχει η συστάδα. Οπότε αποκτούμε έτσι γρηγορά μια ποσοτική αντίληψη του πόσο πυκνά είναι οι στάσεις μας. Το εργαλείο αυτό θα το χρησιμοποιήσουμε κυρίως στο δεύτερο μέρος της ανάλυσης.

3.2 Concorde TSP Solver For Windows

Το Concorde είναι ένας κώδικας υπολογιστή που χρησιμοποιείται για την επίλυση του Συμμετρικού Προβλήματος του Περιοδεύοντος Πωλητή καθώς και ορισμένων άλλων σχετικών προβλημάτων βελτιστοποίησης δικτύου. Ο κώδικας είναι γραμμένος στη γλώσσα προγραμματισμού ANSI C και είναι ελεύθερα διαθέσιμος για ακαδημαϊκή έρευνα. Ο TSP Solver της Concorde έχει χρησιμοποιηθεί για να προσδιοριστούν τα ολικά βέλτιστα και των 110 περιπτώσεων που εμπεριέχονται στην TSPLIB, εκ των οποίων η πολυπλοκότερη περίπτωση έχει 85.900 πόλεις.

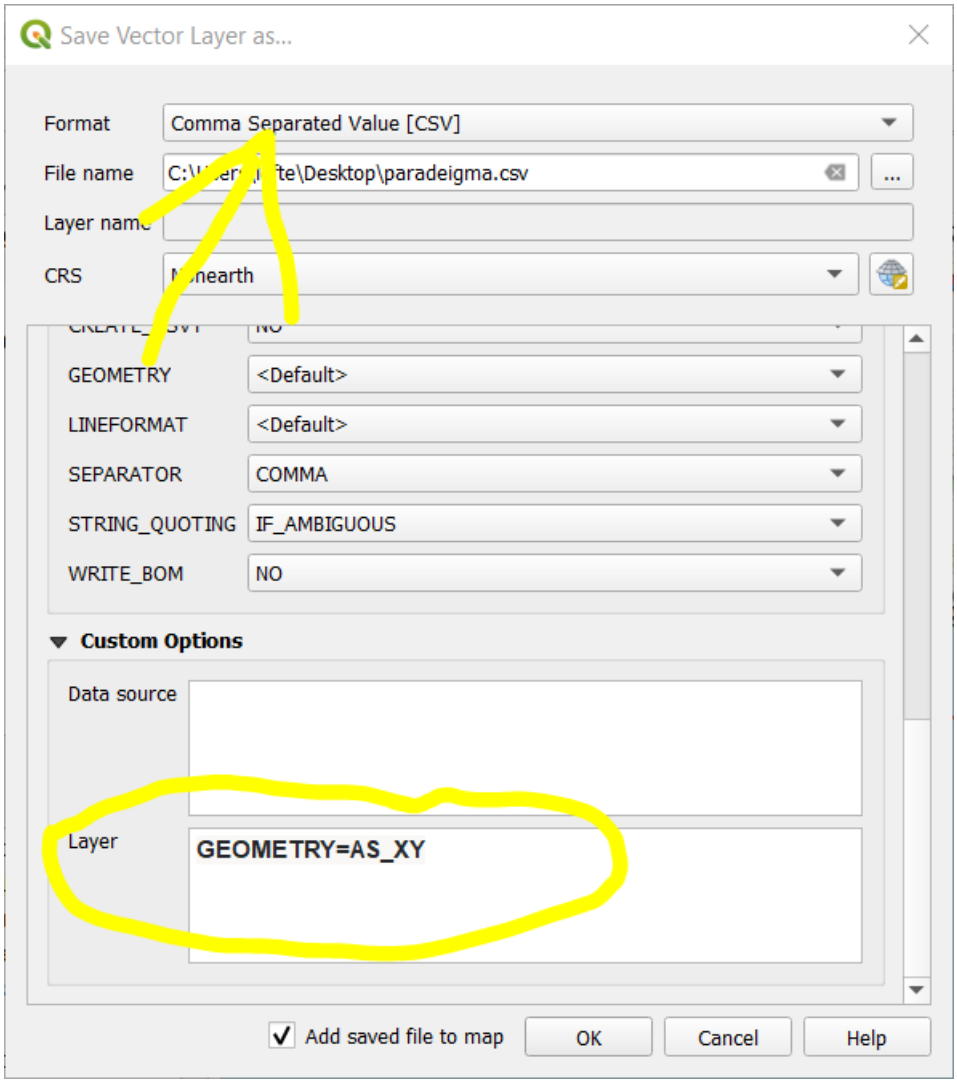
Το γραφικό περιβάλλον του είναι ιδιαίτερα απλό. Για να του εισάγουμε σημεία και να λειτουργήσουμε την ευρετική των Lin – Kernighan που περιέχει, δέχεται ένα ειδικό φορματ αρχείων τα αρχεία .qs , τα οποία είναι αρχεία text (tab delimited).

Θα κάνουμε ένα παράδειγμα εδώ για μια συστάδα από τις 100 που δημιουργήσαμε πριν με το QGIS.



<u>Σχήμα 3.8 Συστάδα για την οποία θα βρούμε την βέλτιστη δρομολόγηση</u>

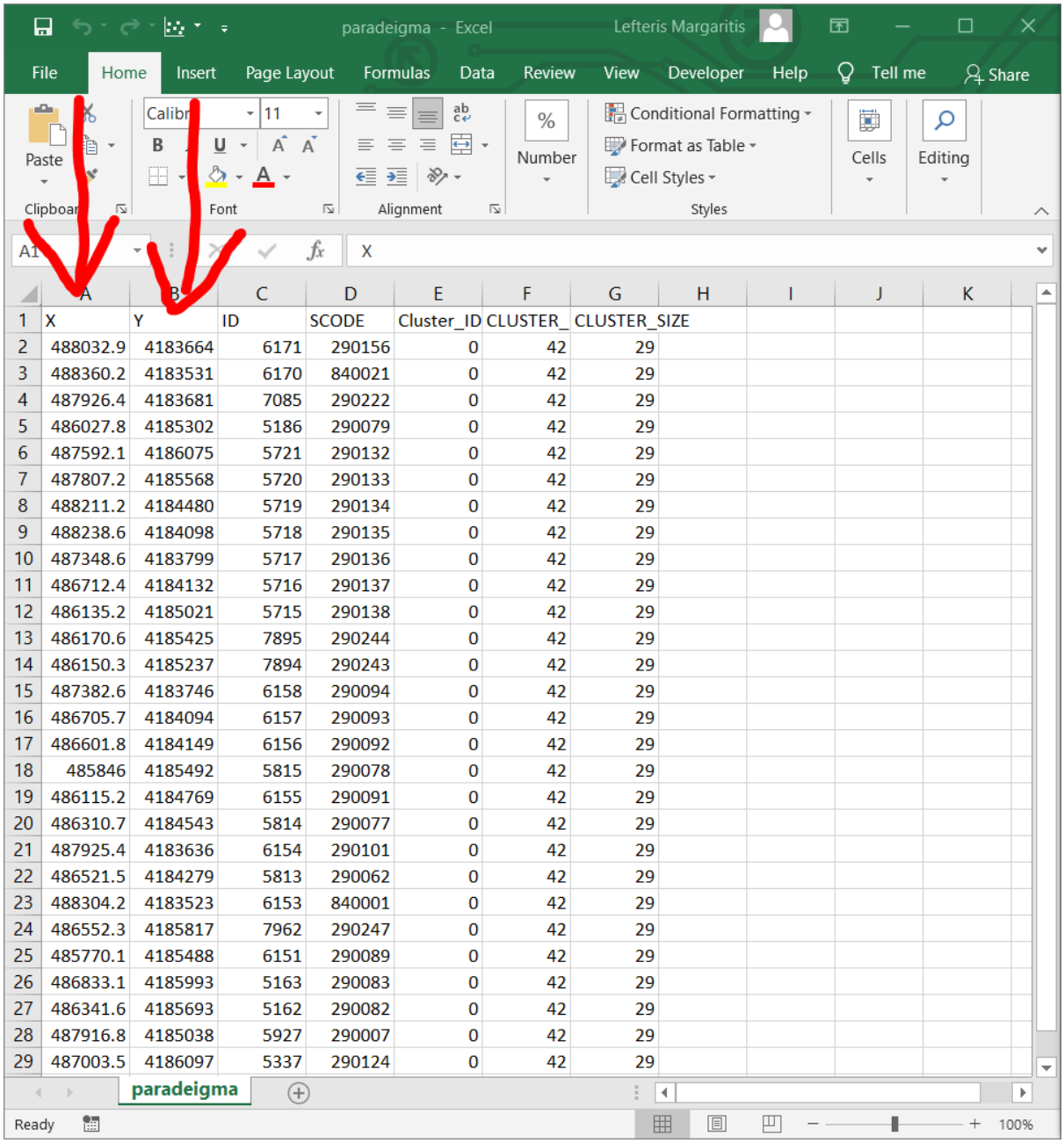
Αφού διαλέξουμε τα σημεία της συστάδας που θέλουμε (στο selection toolbar → select by free hand) κάνουμε δεξί κλικ στο layer Clusters και πατάμε export → save selected features as…



Σχήμα 3.9

Τα αποθηκεύουμε σε αρχείο μορφής csv ώστε να μπορούμε να το επεξεργαστούμε και κάτω από τα Custom Options στο πεδίο Layer συμπληρώνουμε την εντολή **GEOMETRY=AS_XY** προκειμένου να πάρουμε τις γεωγραφικές συντεταγμένες αυτών των σημείων.

Έπειτα ανοίγουμε το αρχείο csv που δημιουργήσαμε:



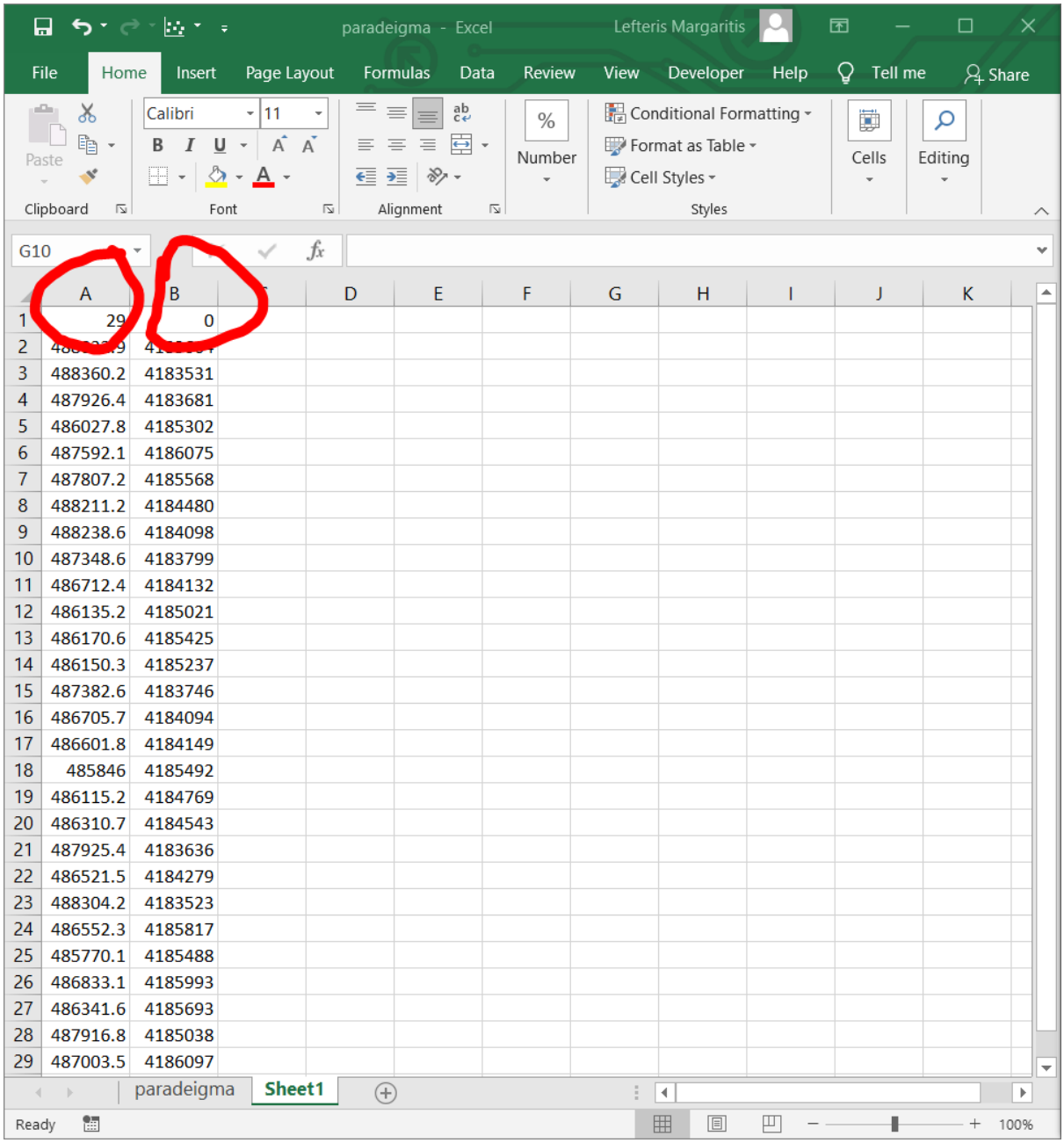
	A	B	C	D	E	F	G
1	X	Y	ID	SCODE	Cluster_ID	CLUSTER_	CLUSTER_SIZE
2	488032.9	4183664	6171	290156	0	42	29
3	488360.2	4183531	6170	840021	0	42	29
4	487926.4	4183681	7085	290222	0	42	29
5	486027.8	4185302	5186	290079	0	42	29
6	487592.1	4186075	5721	290132	0	42	29
7	487807.2	4185568	5720	290133	0	42	29
8	488211.2	4184480	5719	290134	0	42	29
9	488238.6	4184098	5718	290135	0	42	29
10	487348.6	4183799	5717	290136	0	42	29
11	486712.4	4184132	5716	290137	0	42	29
12	486135.2	4185021	5715	290138	0	42	29
13	486170.6	4185425	7895	290244	0	42	29
14	486150.3	4185237	7894	290243	0	42	29
15	487382.6	4183746	6158	290094	0	42	29
16	486705.7	4184094	6157	290093	0	42	29
17	486601.8	4184149	6156	290092	0	42	29
18	485846	4185492	5815	290078	0	42	29
19	486115.2	4184769	6155	290091	0	42	29
20	486310.7	4184543	5814	290077	0	42	29
21	487925.4	4183636	6154	290101	0	42	29
22	486521.5	4184279	5813	290062	0	42	29
23	488304.2	4183523	6153	840001	0	42	29
24	486552.3	4185817	7962	290247	0	42	29
25	485770.1	4185488	6151	290089	0	42	29
26	486833.1	4185993	5163	290083	0	42	29
27	486341.6	4185693	5162	290082	0	42	29
28	487916.8	4185038	5927	290007	0	42	29
29	487003.5	4186097	5337	290124	0	42	29

Σχήμα 3.10

Και βλέπουμε ότι στις δυο πρώτες στήλες έχουμε τις γεωγραφικές συντεταγμένες των σημείων της συστάδας μας.

Για να δημιουργήσουμε ένα λειτουργικό αρχείο .qs πρέπει να ακολουθήσουμε ΑΥΣΤΗΡΑ τα εξής βήματα:

- Απομονώνουμε τις στήλες με τις συντεταγμένες και στην πρώτη στήλη της πρώτης γραμμής εισάγουμε το σύνολο των σημείων που έχουμε, στην περίπτωση μας τον αριθμό 29 και στην δεύτερη στήλη της πρώτης γραμμής εισάγουμε το ειδικό βάρος που θα ενωθούν οι κόμβοι, στην περίπτωση μας 0. Γενικά όμως στην συνέχεια της διπλωματικής θα αυξήσουμε το βάρος της διαδρομής με έναν συντελεστή 1,5 για να προσομοιάσουμε πιο ρεαλιστικές συνθήκες οδικού δικτύου και όχι ευκλείδειας γεωμετρίας.

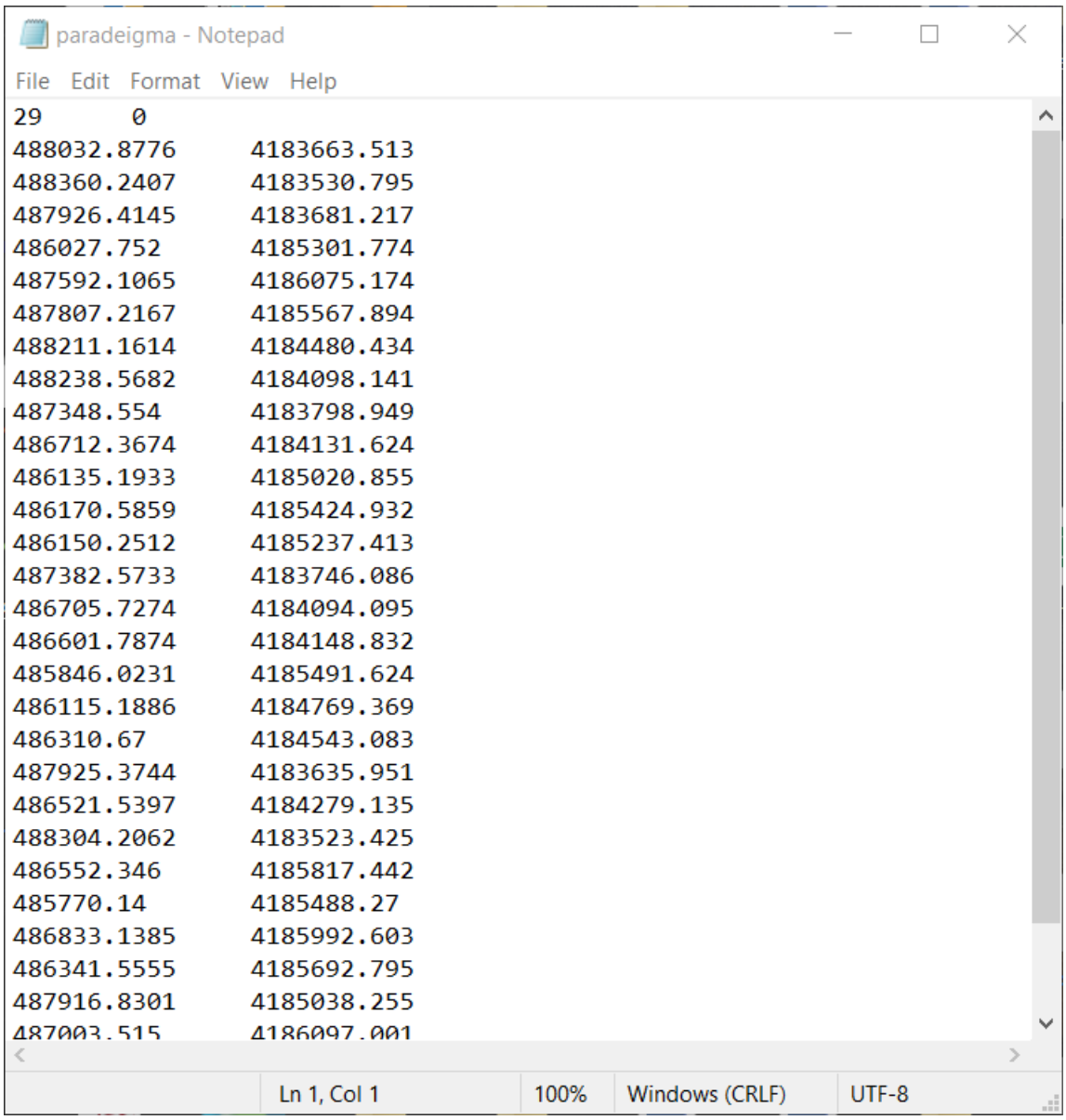


	A	B
1	29	0
2	[illegible]	[illegible]
3	488360.2	4183531
4	487926.4	4183681
5	486027.8	4185302
6	487592.1	4186075
7	487807.2	4185568
8	488211.2	4184480
9	488238.6	4184098
10	487348.6	4183799
11	486712.4	4184132
12	486135.2	4185021
13	486170.6	4185425
14	486150.3	4185237
15	487382.6	4183746
16	486705.7	4184094
17	486601.8	4184149
18	485846	4185492
19	486115.2	4184769
20	486310.7	4184543
21	487925.4	4183636
22	486521.5	4184279
23	488304.2	4183523
24	486552.3	4185817
25	485770.1	4185488
26	486833.1	4185993
27	486341.6	4185693
28	487916.8	4185038
29	487003.5	4186097

Σχήμα 3.11

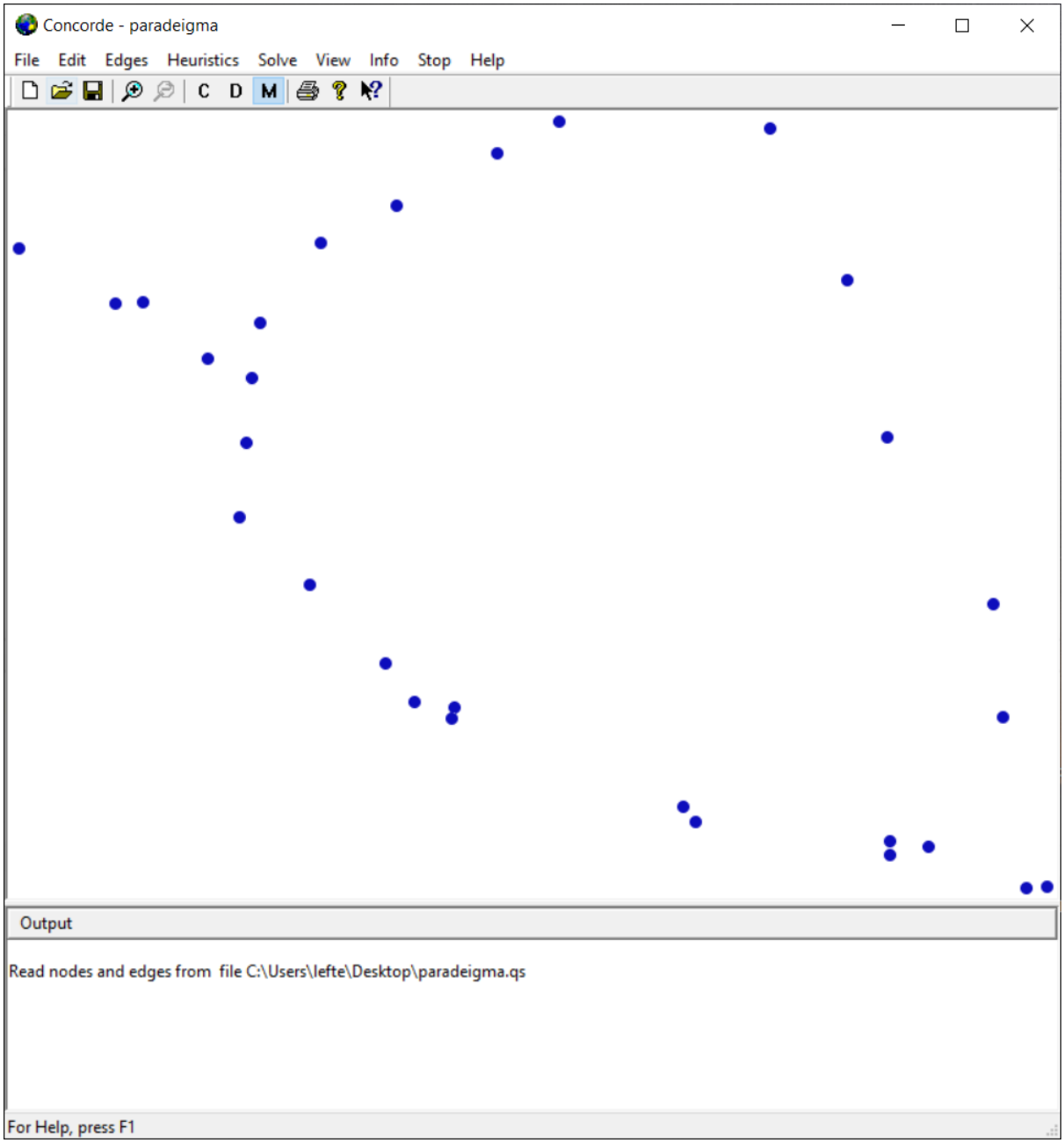
Έπειτα αποθηκεύουμε το αρχείο ως text(tab delimited) και αφού το ανοίξουμε με το notepad πάμε file →save as και στο save as type διαλέγουμε all files και στον τίτλο προσθέτουμε την κατάληξη .qs

Τελικά το αρχείο μας θα πρέπει να έχει αυτή την μορφή:



Σχήμα 3.12

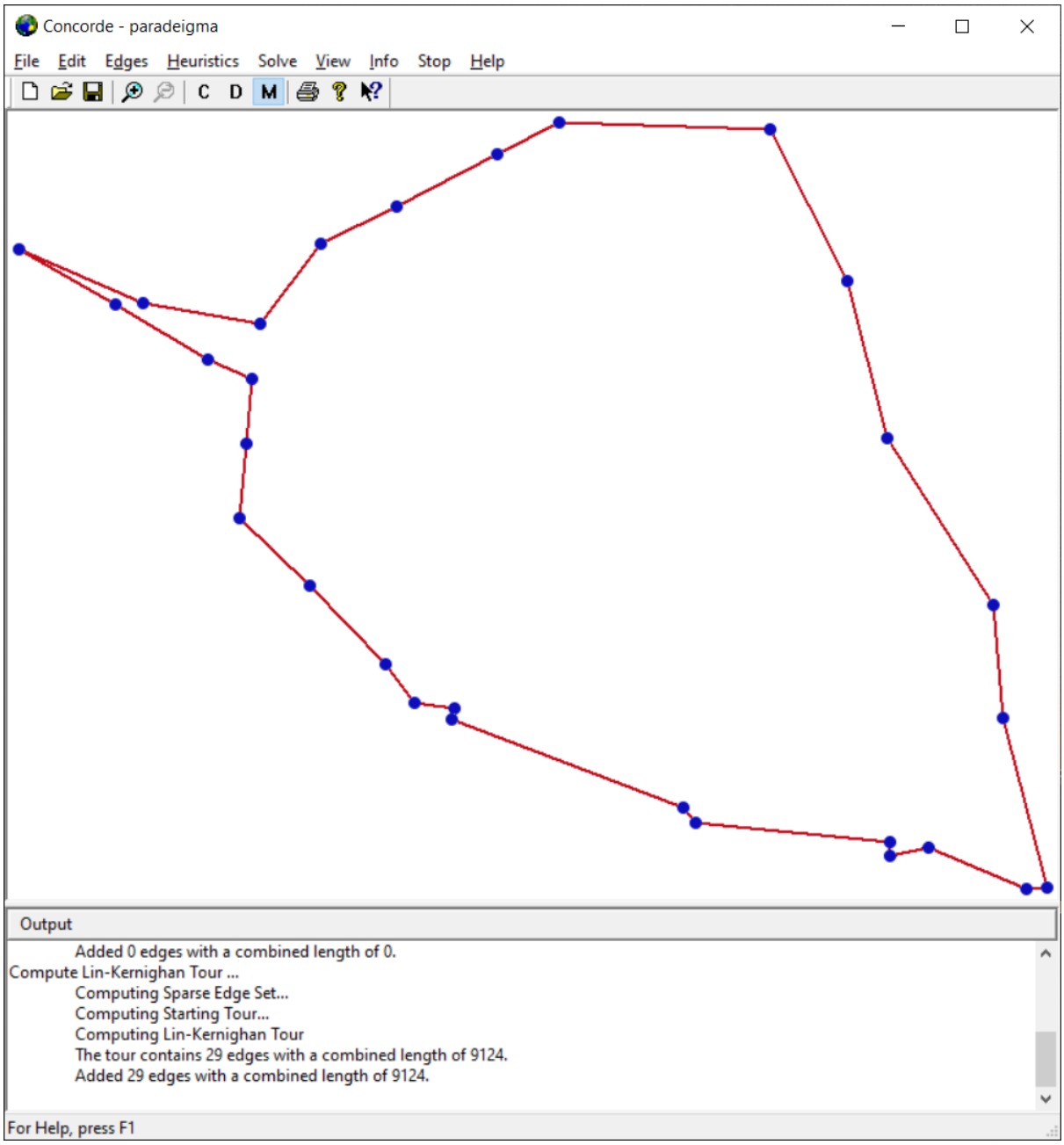
Και έπειτα μπορούμε να το ανοίξουμε με τον Concorde TSP Solver.



Σχήμα 3.13

Βλέπουμε λοιπόν την διάταξη των σημείων μας. Και για να κάνουμε χρήση του αλγορίθμου Lin – Kernighan πάμε στην επιλογή Heuristics → Lin Kernighan χρησιμοποιούμαι τις default ρυθμίσεις στο παράθυρο που μας πετάει και πατάμε OK.

Στην συνέχεια παίρνουμε τα αποτελέσματα:



Σχήμα 3.14

Από όπου μπορούμε να δούμε ότι η βέλτιστη διαδρομή που βρήκε ο αλγόριθμος είναι 9124 μέτρα.

4. Αντιμετώπιση προβλήματος με δημιουργία επίπεδων συσταδιοποίησης

4.1 Παράμετροι και υποθέσεις του προβλήματος

Για το πρόβλημα μας, την επιθεώρηση του δικτύου ΟΑΣΑ θα λάβουμε υπόψιν δυο παραμέτρους, την ταχύτητα με την οποία κινείται το όχημα επιθεώρησης και την διάρκεια που χρειάζεται ο επιθεωρητής σε κάθε στάση ελέγχου για να σημειώσει τις παρατηρήσεις του.

Στόχος μας είναι αρχικά να υποδιαιρέσουμε το σύνολο του δικτύου σε μικρότερες ομάδες, τις λεγόμενες και συστάδες, ώστε το σύνολο της διάρκειας της επιθεώρησης να κρατάει όσο το δυνατόν πιο κοντά γίνεται σε 8 ώρες, δηλαδή την διάρκεια μιας ημέρας πλήρους απασχόλησης ενός εργαζομένου.

Το γεγονός αυτό αυξάνει ακόμα περισσότερο την πολυπλοκότητα του προβλήματος, διότι πέρα από τα δυο προβλήματα εξαιρετικάς μεγάλης πολυπλοκότητας, της εύρεσης βέλτιστης δρομολόγησης και της ομοιόμορφης ομαδοποίησης προστίθεται και η διάσταση του χρόνου στο πρόβλημα.

Για το λόγο αυτό η στρατηγική μας στην αντιμετώπιση θα επικεντρωθεί αρχικά στο όσο το δυνατόν πιο καλή υποδιαίρεση του δικτύου ΟΑΣΑ. Για την διαδικασία αυτή θα χρησιμοποιήσουμε το πρόγραμμα QGIS και το εργαλείο ανάλυσης που διαθέτει με το όνομα k-means clustering.

Ωστόσο η ομαδοποίηση με τον αλγόριθμο K – means ενώ είναι εξαιρετικά χρήσιμη στην περίπτωση μας επειδή εξασφαλίζει ότι ο τρόπος που θα δημιουργήσει τις συστάδες θα είναι τέτοιος ώστε οι στάσεις να είναι όσο το δυνατόν πιο κοντά η μια στην άλλη (μαζεμένες), έχει το μειονέκτημα ότι δεν έχουμε απολυτό έλεγχο στον αριθμό των στάσεων που θα έχει η κάθε συστάδα. Για τον λόγο αυτό θα ξεκινήσουμε δημιουργώντας μεγάλες συστάδες, οι οποίες θα είναι ανέφικτο να επιτηρηθούν σε 8 ώρες και μετά θα ξαναπαίρναμε σε επόμενη συσταδιοποιήση ανάλογα με το πόσο υπερβαίνει η επιτήρηση τους τις 8 ώρες. Είναι ιδιαίτερα σημαντικό αυτού του είδους η διαδικασία να γίνει σε πολλαπλά βήματα ανάλυσης, ώστε όλες οι ομαδοποιήσεις που θα προκύψουν να μπορούν να αποτελούν μια καλή λύση στο πρόβλημα μας. Κάθε βήμα επιπλέον υποδιαίρεσης του δικτύου θα το υποδιαίρει το δίκτυο το πολύ σε 5 επιπλέον συστάδες.

Το πρόβλημα μας θα λυθεί για δυο διαφορετικά ζεύγη παραμέτρων όσων αφορά την χρονική του διάσταση τα οποία είναι τα εξής:

- Ταχύτητα οχήματος επιθεώρησης τα 10km/h και χρόνος διάρκειας για επιθεώρηση στάσης ίσος με δυο λεπτά
  Την οποία ονομάζουμε **γρήγορη επίλυση**.
- Ταχύτητα οχήματος επιθεώρησης τα 5km/h και χρόνος διάρκειας για επιθεώρηση στάσης ίσος με ένα λεπτό.
  Την οποία ονομάζουμε **αργή επίλυση**.

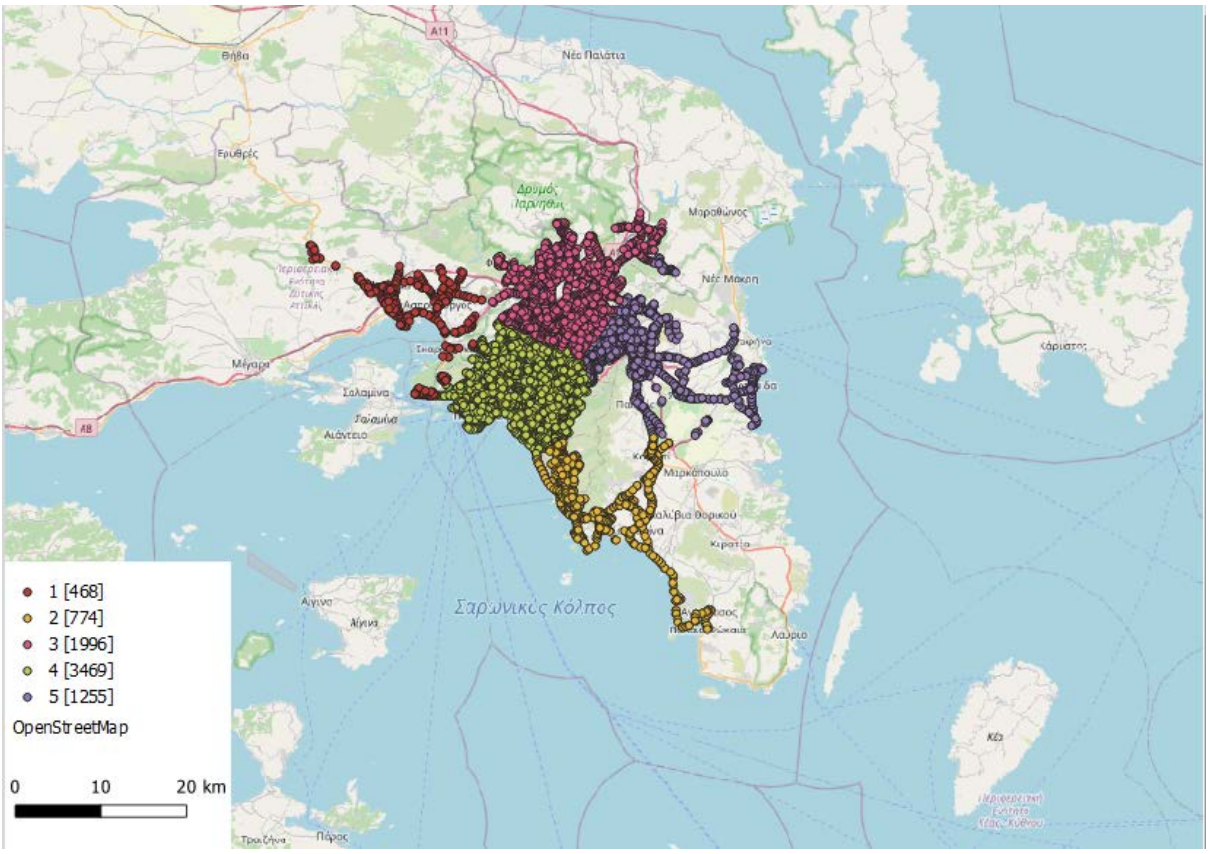
Παρατηρούμε ότι τα ζεύγη που διαλέξαμε συνδυάζουν μεγάλη ταχύτητα με μικρή διάρκεια στάσης και αντίστροφα, η στρατηγική μας είναι τέτοια ώστε να βρούμε μεγάλα και μικρά όρια ώστε μετά να είναι ευκολά προσαρμόσιμα σε αληθινές συνθήκες επιθεώρησης, έπειτα από ένα μικρό δείγμα για τους χρόνους που χρειάζεται ο επιθεωρητής.

Μια τελική παρατήρηση η οποία είναι και απολυτά χρήσιμη είναι το γεγονός ότι η ευρετική μέθοδος Lin-Kernighan της οποίας θα κάνουμε χρήση για να βρούμε την βέλτιστη δρομολόγηση, την δουλεύουμε σε περιβάλλον ευκλείδειας γεωμετρίας καθώς δεν έχουμε δεδομένα για την απόσταση των στάσεων μέσω της χρήσης του οδικού δικτύου. Για αυτό τον λόγο θα αυξήσουμε το τελικό μας μήκος διαδρομής με έναν συντελεστή α = 1.5 για να προσπαθήσουμε να προσομοιάσουμε πραγματικές συνθήκες.

4.2 Πρώτο επίπεδο ανάλυσης

Αρχικά υποδιαιρέσαμε το δίκτυο σε πέντε ομάδες με χρήση του αλγορίθμου k – means clustering.

Με input layer το σύνολο του δικτύου stasis (7962 στοιχεία) και επιλογή δημιουργίας 5 cluster όπως είδαμε και στο 3ο κεφάλαιο παίρνουμε τον εξής χάρτη:

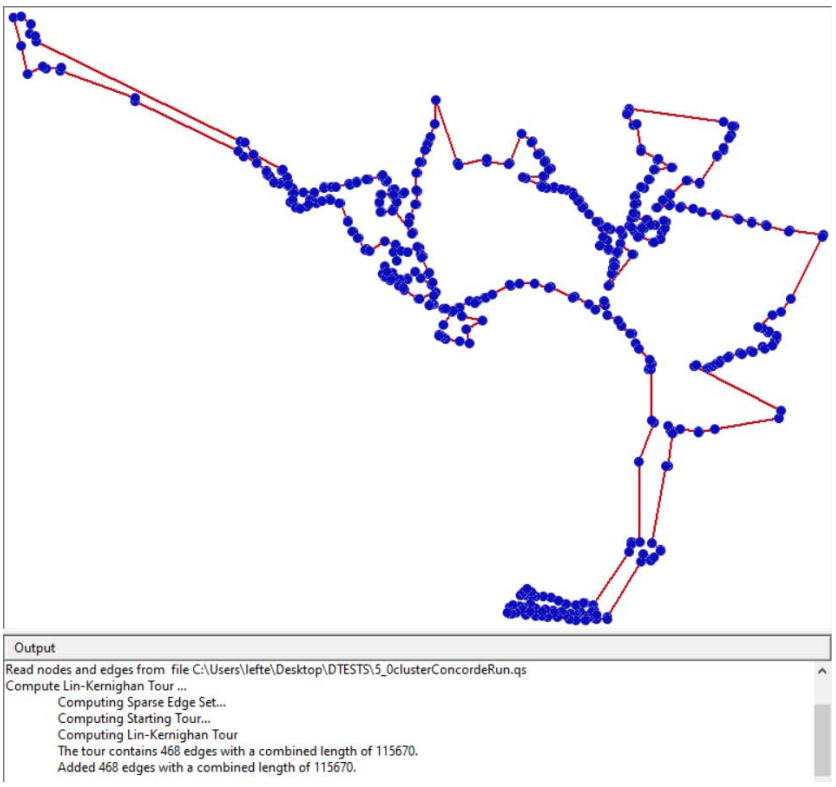


Σχήμα 4.1 Υποδιαίρεση του δικτύου σε 5 συστάδες

Όπως βλέπουμε σε αύξουσα σειρά οι συστάδες μας έχουν 468 / 774 / 1255 / 1996 / 3469 στάσεις του δικτύου ΟΑΣΑ.

Αναλύουμε για την κάθε συστάδα πόση ώρα θα έπαιρνε η επιθεώρηση της. Με χρήση της ευχετικής μεθόδου Lin-Kernighan μέσω του λογισμικού Concorde παίρνουμε τα εξής αποτελέσματα:

Συστάδα 468 στάσεων:

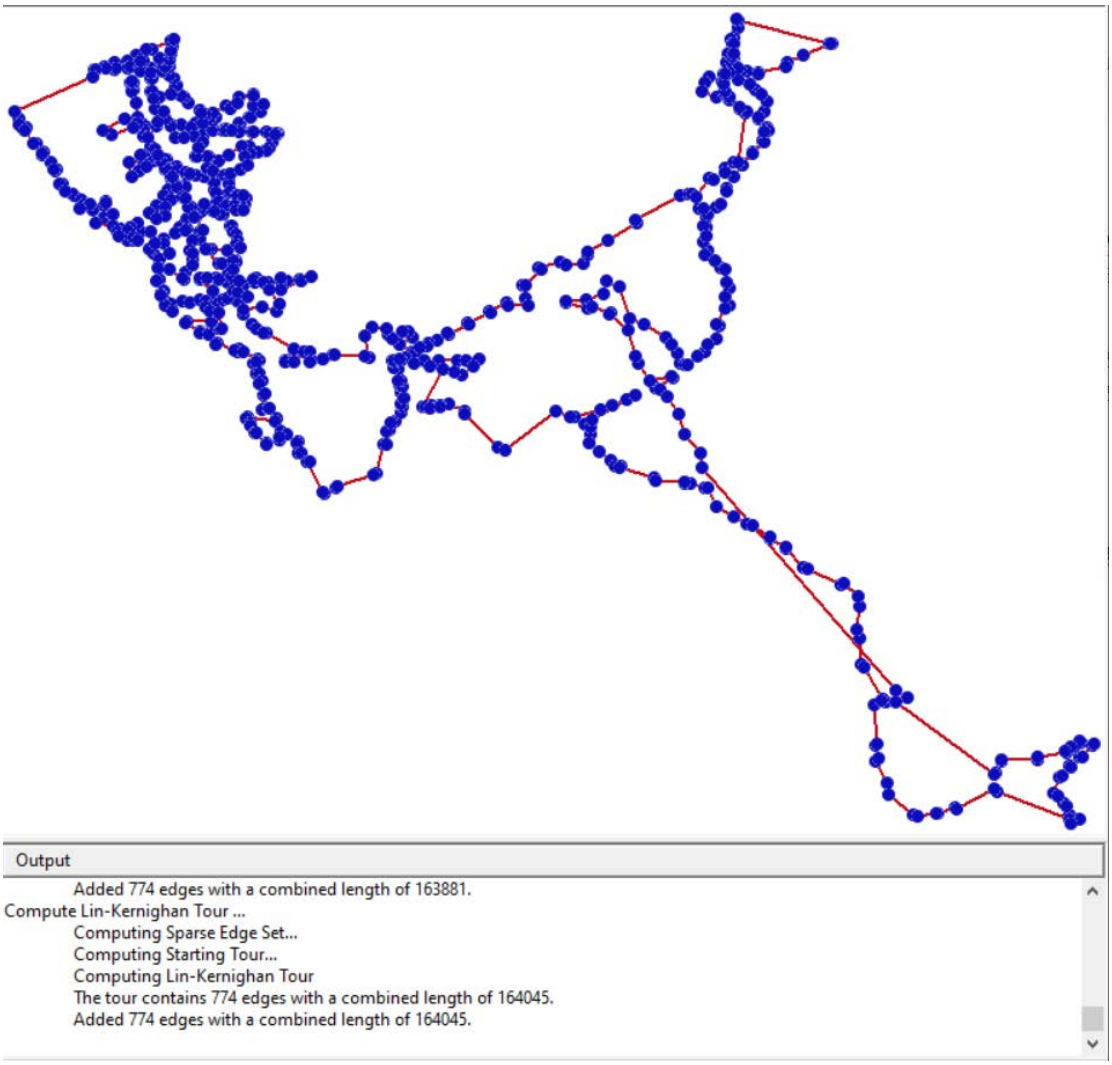


Σχήμα 4.2 Βέλτιστη δρομολόγηση για την 1η συστάδα

Βλέπουμε ότι το τελικό μήκος διαδρομής που πρέπει να διανθεί για αυτή την συστάδα είναι 115670m*1.5 ή 173,5 χιλιόμετρα, οπότε τελική χρονική διάρκεια ίση με:

- 173,5/10 + 468*1/60 = 25,15 ώρες για γρήγορη επίλυση
- 173,5/5 + 468*2/60 = 50,3 ώρες για αργή επίλυση

Συστάδα 774 στάσεων:

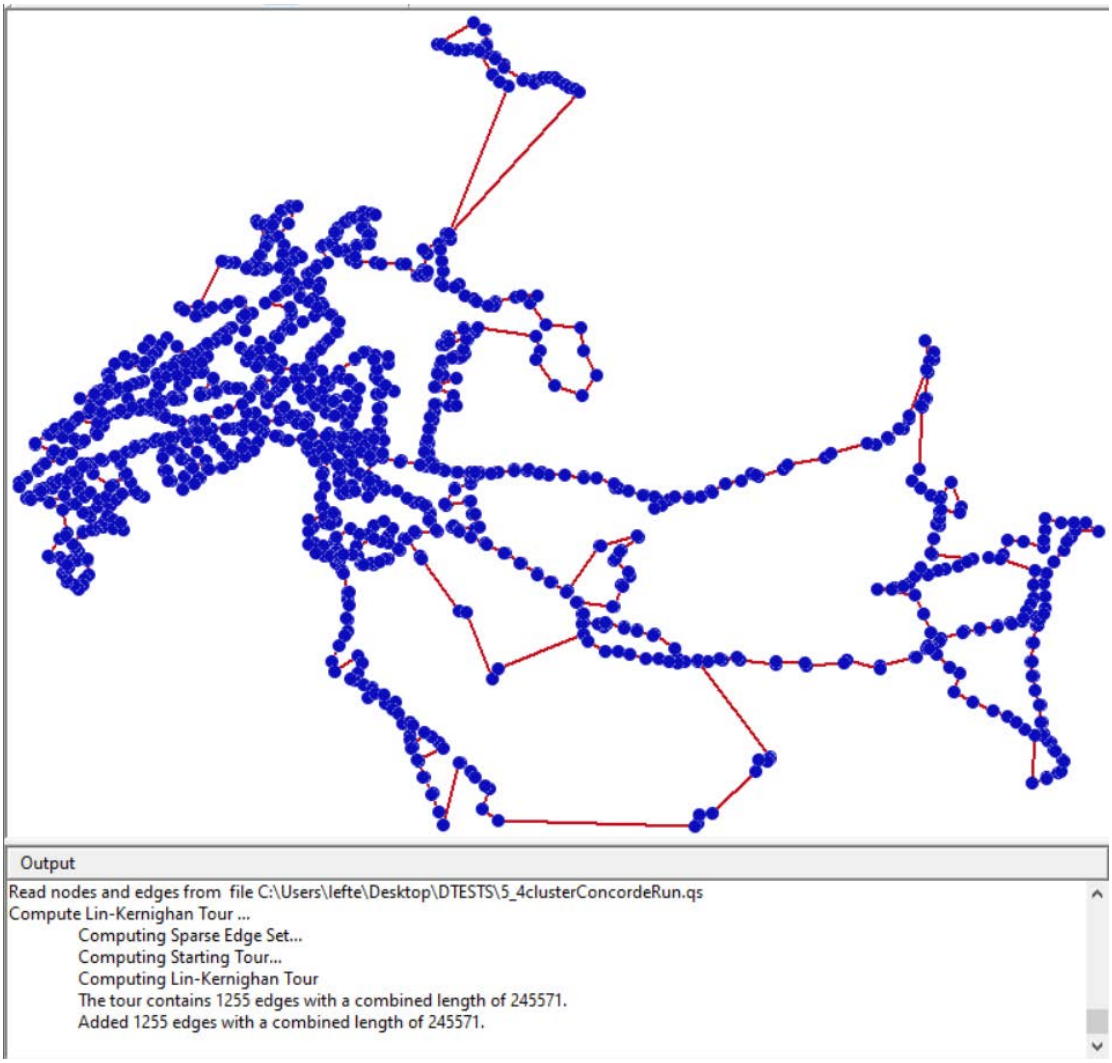


Σχήμα 4.3 Βέλτιστη δρομολόγηση για την 2η συστάδα

Βλέπουμε ότι το τελικό μήκος διαδρομής που πρέπει να διανθεί για αυτή την συστάδα είναι 164045m*1.5 ή 246.1 χιλιόμετρα, οπότε τελική χρονική διάρκεια ίση με:

- 246.1/10 + 774*1/60 = 37.51 ώρες για γρήγορη επίλυση
- 246.1/5 + 774*2/60 = 75.02 ώρες για αργή επίλυση

Συστάδα 1255 στάσεων:

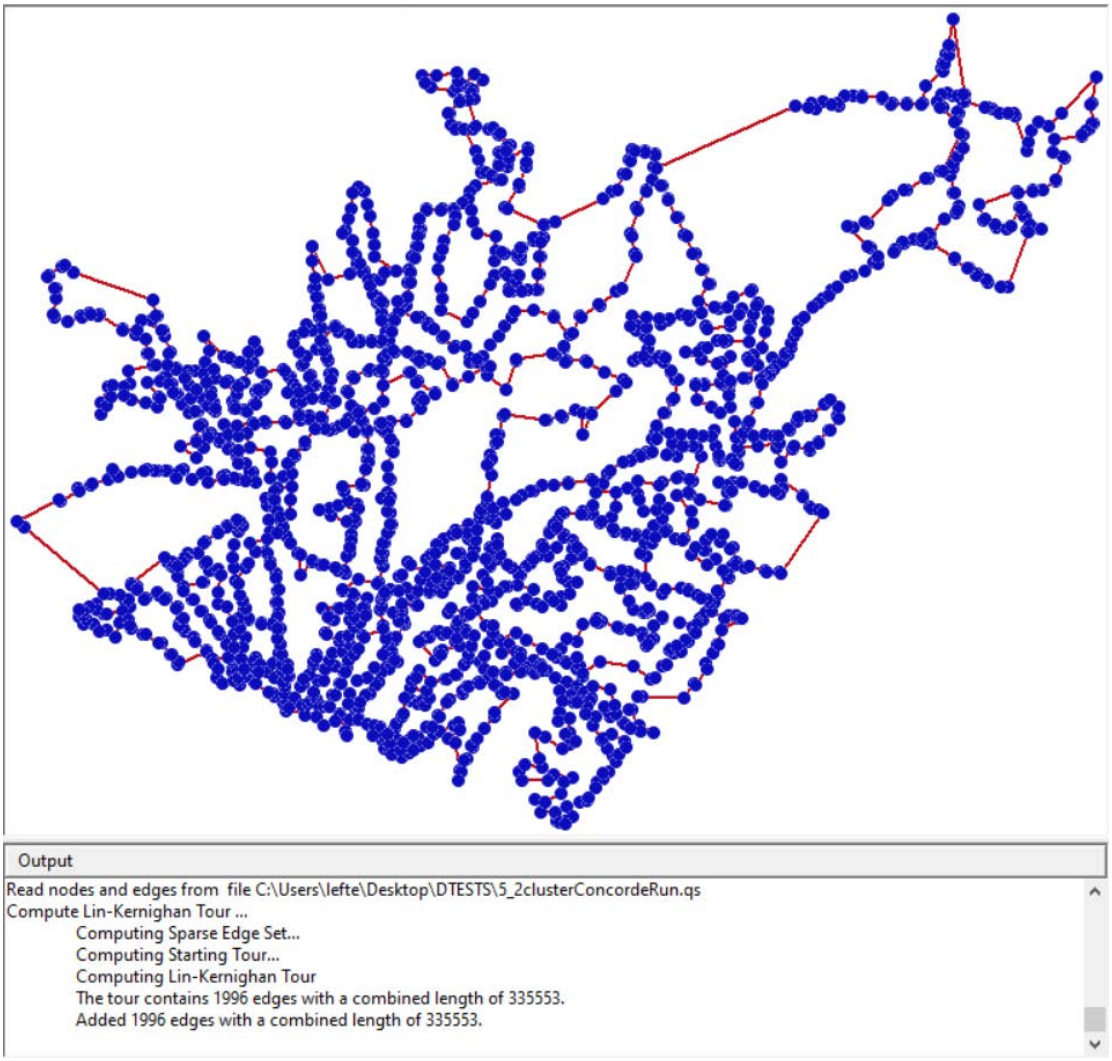


Σχήμα 4.4 Βέλτιστη δρομολόγηση για την 3η συστάδα

Βλέπουμε ότι το τελικό μήκος διαδρομής που πρέπει να διανθεί για αυτή την συστάδα είναι 245571m*1.5 ή 368.4 χιλιόμετρα, οπότε τελική χρονική διάρκεια ίση με:

- 368.4/10 + 1255*1/60 = 57.76 ώρες για γρήγορη επίλυση
- 368.4/5 + 1255*2/60 = 115.51 ώρες για αργή επίλυση

Συστάδα 1996 στάσεων:

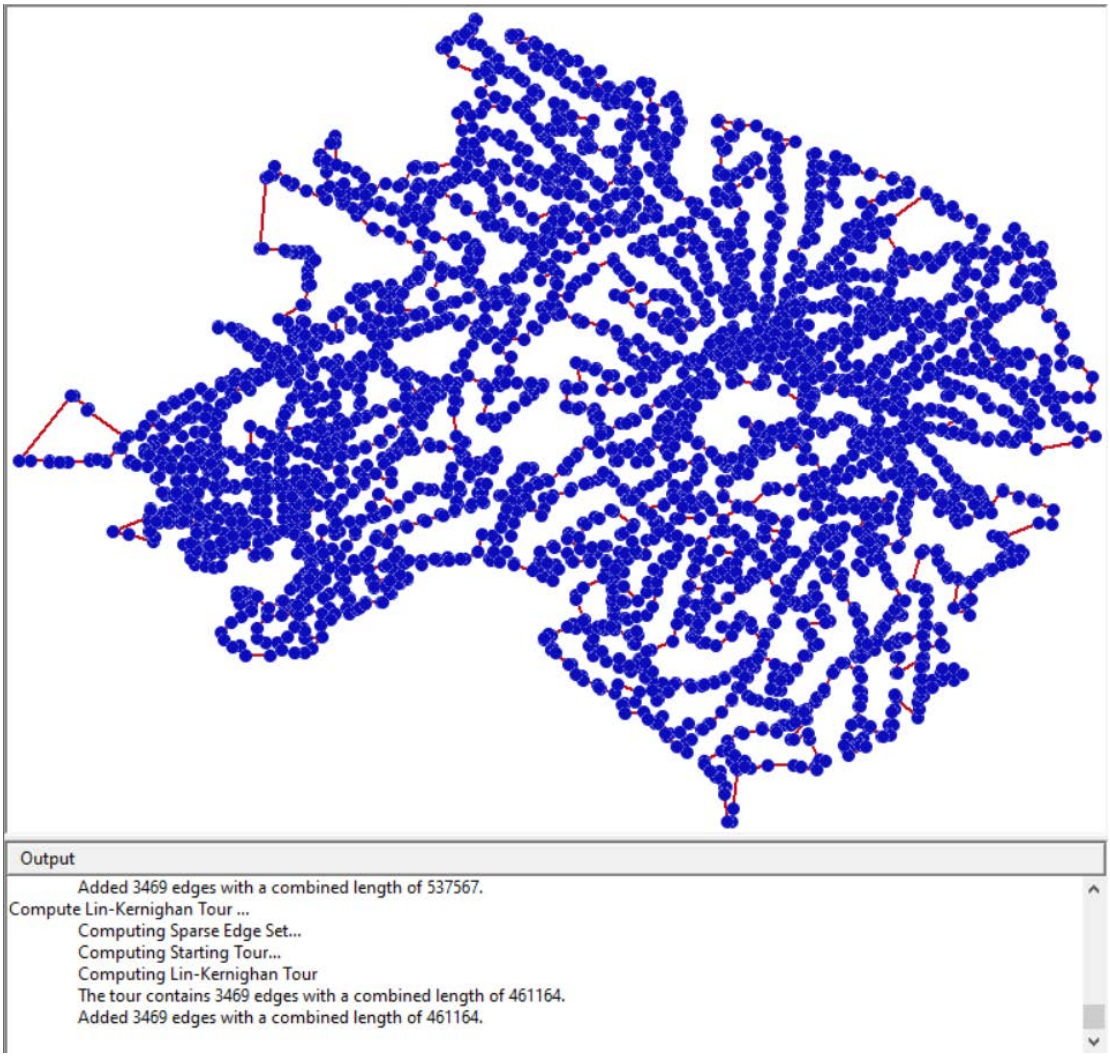


Σχήμα 4.5 Βέλτιστη δρομολόγηση για την 4η συστάδα

Βλέπουμε ότι το τελικό μήκος διαδρομής που πρέπει να διανθεί για αυτή την συστάδα είναι 335553m*1.5 ή 503.3 χιλιόμετρα, οπότε τελική χρονική διάρκεια ίση με:

- 503.3/10 + 1996*1/60 = 83.6 ώρες για γρήγορη επίλυση
- 503.3/5 + 1996*2/60 = 167.2 ώρες για αργή επίλυση

Συστάδα 3469 στάσεων:



Σχήμα 4.6 Βέλτιστη δρομολόγηση για την 5η συστάδα

Βλέπουμε ότι το τελικό μήκος διαδρομής που πρέπει να διανθεί για αυτή την συστάδα είναι 461164m*1.5 ή 691.7 χιλιόμετρα, οπότε τελική χρονική διάρκεια ίση με:

- 691.7/10 + 3469*1/60 = 127 ώρες για γρήγορη επίλυση
- 691.7/5 + 3469*2/60 = 254 ώρες για αργή επίλυση

Τελικώς από το πρώτο στάδιο ανάλυσης έχουμε τα αποτελέσματα:

Πίνακας 4.1 Αποτελέσματα των 5 συστάδων

Cluster Size	Minimum Distance(m)	Time(fast) in hours	Time(slow) in hours
468	115670	25.1505	50.301
774	164045	37.50675	75.0135
1255	245571	57.75231667	115.5046333
1996	335553	83.59961667	167.1992333
3469	461164	126.9912667	253.9825333

4.3 Δεύτερο επίπεδο ανάλυσης

Για το δεύτερο στάδιο ανάλυσης, θα αρχίσουμε από την περαιτέρω υποδιαίρεση βάση των χρονών της γρήγορης επίλυσης και όταν ολοκληρωθεί η ανάλυση της γρήγορης επίλυσης θα περάσουμε στην αργή επίλυση.

Επειδή ο μέγιστος αριθμός υποδιαίρεσης είναι το 5, θα διαιρέσουμε τον απαιτούμενο χρόνο της γρήγορης επίλυσης με το 8, που είναι οι μέγιστες ώρες και θα στρογγυλοποιήσουμε με μέγιστο αριθμό το 5, για να βρούμε σε τι περαιτέρω υποδιαίρεση θα συνεχίσουμε.

Οπότε έχουμε όλες οι συστάδες μας θα υποδιαιρεθούν κατά 5 εκτός της πρώτης που θα υποδιαιρεθεί κατά 3.

- **Ανάλυση πρώτης συστάδας 468 στάσεων:**

Με την ιδιά διαδικασία που είδαμε προχωράμε στην περαιτέρω ανάλυση της συστάδας.

<u>Σχήμα 4.7 Δεύτερο επίπεδο ανάλυσης της 1ης συστάδας</u>

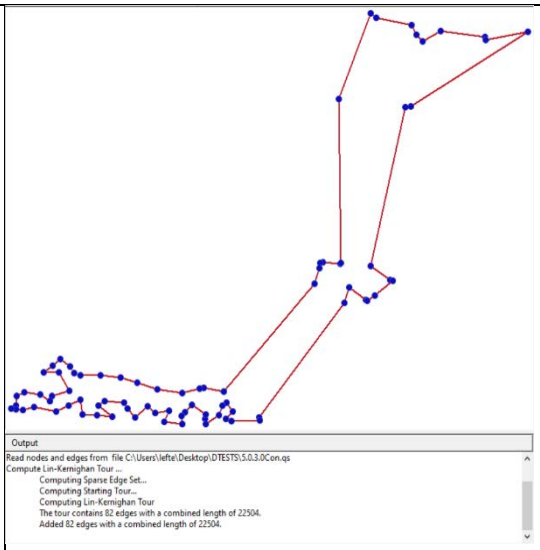
Την υποδιαιρούμε σε τρεις υποσυστάδες μεγέθους σε αύξουσα σειρά 82, 189, 197 στάσεων η κάθε μια.

Από την ανάλυση για βέλτιστη διαδρομή μέσω του αλγορίθμου Lin-Kernighan προκύπτουν τα εξής αποτελέσματα:

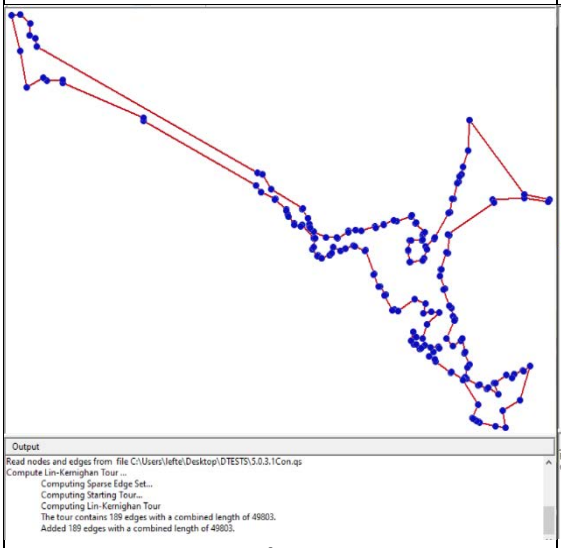
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

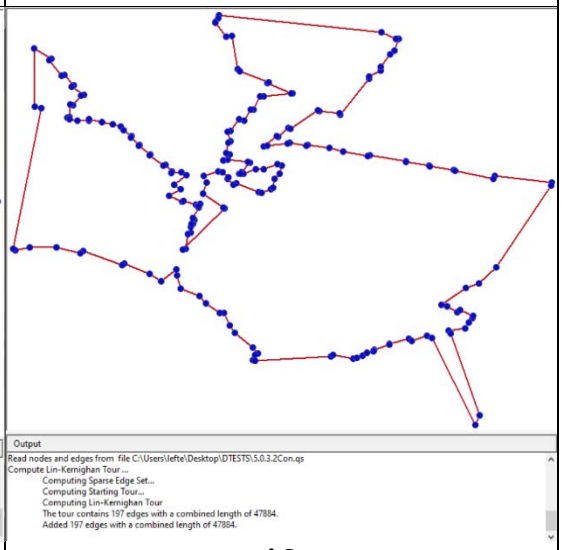
Σειρά: 1.1 / 1.2 / 1.3



Συστάδα 1.1



Συστάδα 1.2



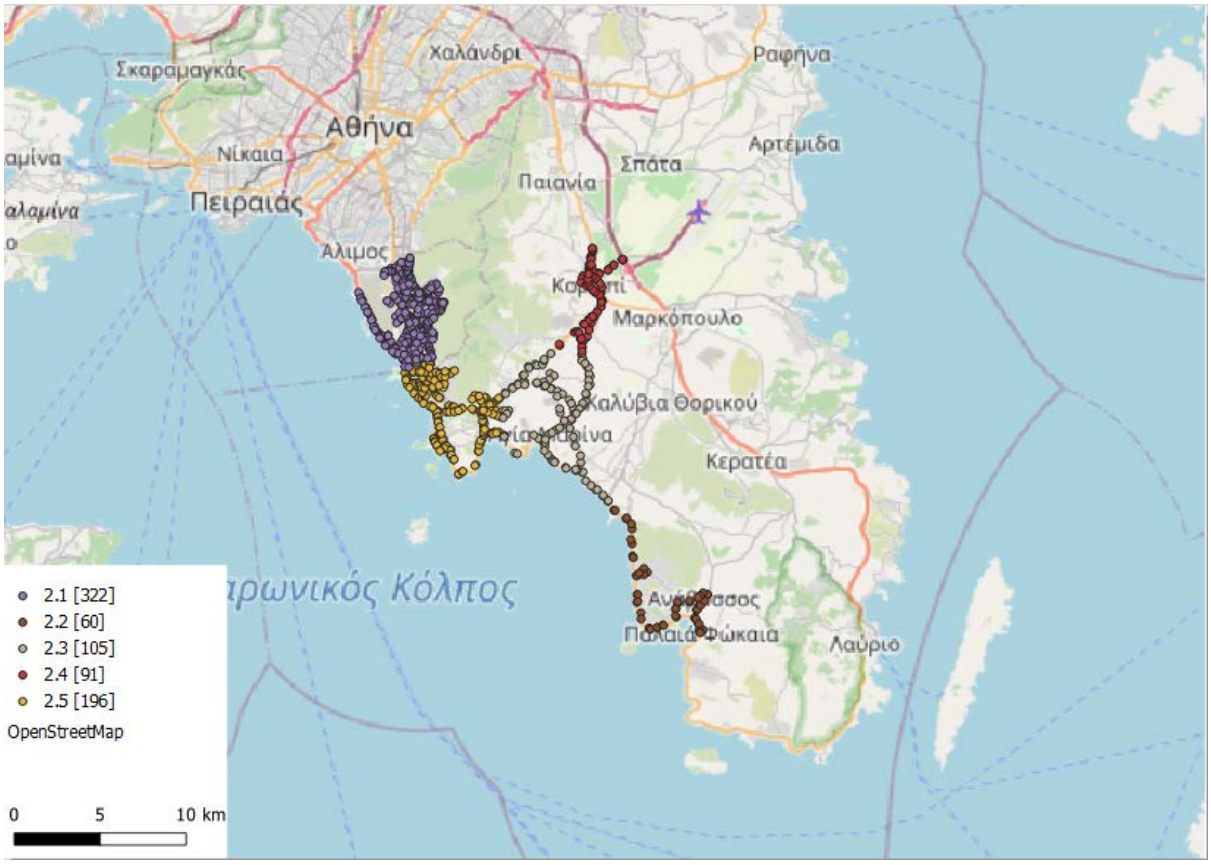
Συστάδα 1.3

Και στον παρακάτω πίνακα βλέπουμε την χρονική ανάλυση:

<u>Πίνακας 4.2 Αποτελέσματα Συστάδων 1.1,1.2,1.3</u>

Cluster Size	Minimum Distance(m)	Time(fast) in hours	Time(slow) in hours
82	22504	4.742266667	9.484533333
189	49803	10.62045	21.2409
197	47884	10.46593333	20.93186667

- **Ανάλυση δεύτερης συστάδας 774 στάσεων:**



<u>Σχήμα 4.8 Δεύτερο επίπεδο ανάλυσης της 2ης συστάδας</u>

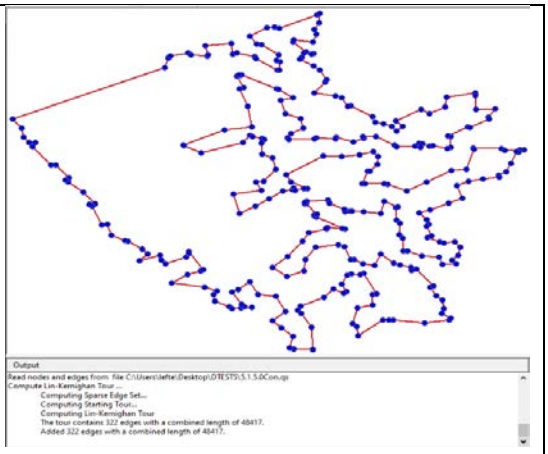
Υποδιαιρούμε την δεύτερη συστάδα μας σε υποσυστάδες μεγέθους 322, 60, 105, 91, 196 στάσεων.

Από την ανάλυση για βέλτιστη διαδρομή μέσω του αλγορίθμου Lin-Kernighan προκύπτουν τα εξής αποτελέσματα:

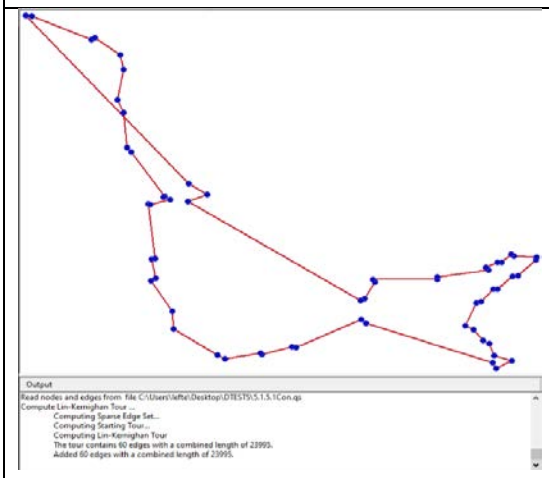
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

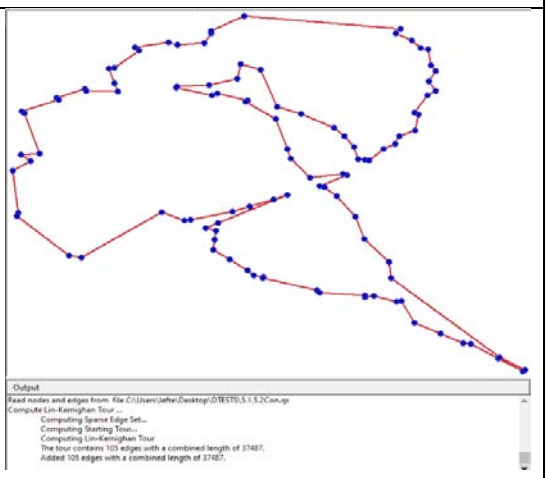
Σειρά: 2.1 / 2.2 / 2.3 / 2.4 / 2.5



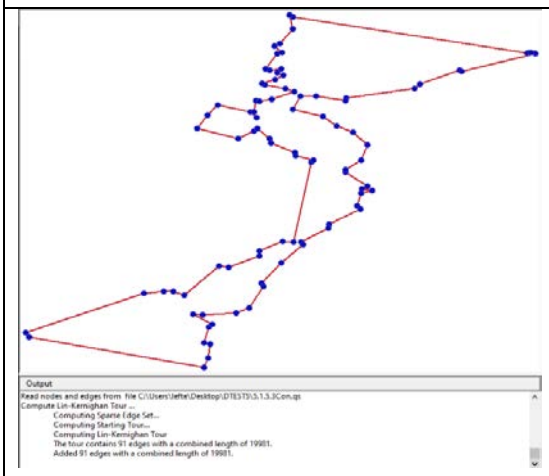
Συστάδα 2.1



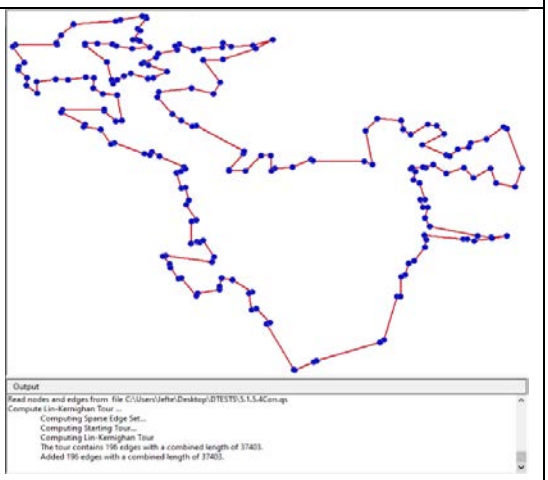
Συστάδα 2.2



Συστάδα 2.3



Συστάδα 2.4



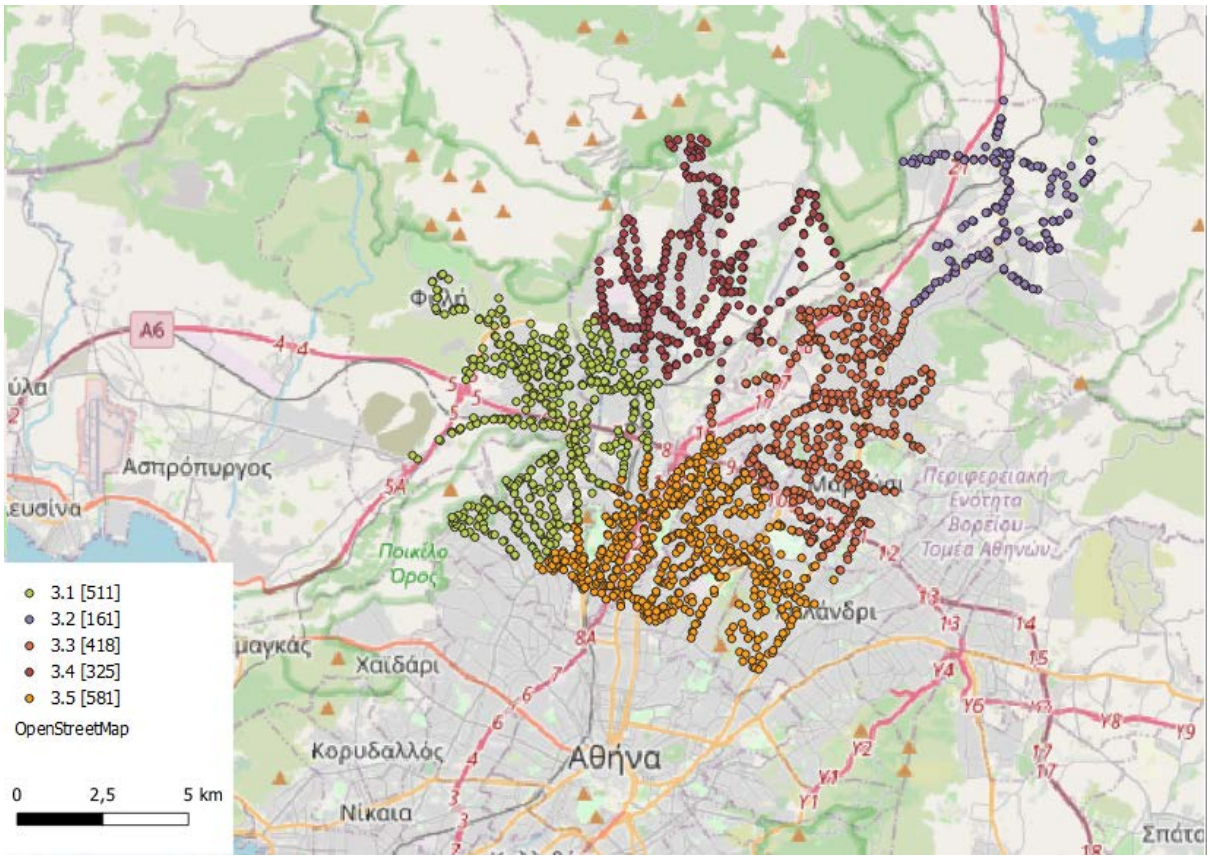
Συστάδα 2.5

Και στον παρακάτω πίνακα βλέπουμε την χρονική ανάλυση:

Πίνακας 4.3 αποτελέσματα συστάδων 2.1,2.2,2.3,2.4,2.5

Cluster Size	Minimum Distance(m)	Time(fast) in hours	Time(slow) in hours
60	23995	4.59925	9.1985
91	19981	4.513816667	9.027633333
105	37487	7.37305	14.7461
196	37403	8.877116667	17.75423333
322	48417	12.62921667	25.25843333

- **Ανάλυση τρίτης συστάδας 1996 στάσεων:**



Σχήμα 4.9 Δεύτερο επίπεδο ανάλυσης της 3ης συστάδας

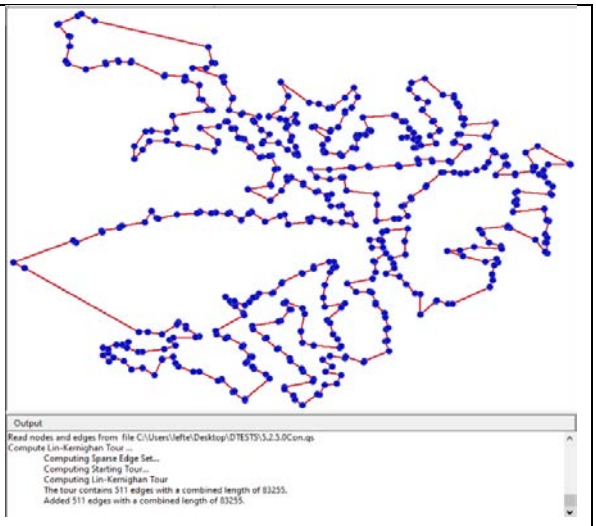
Υποδιαιρούμε την τρίτη συστάδα μας σε υποσυσταδες μεγέθους 511, 161, 418, 325, 581 στάσεων.

Από την ανάλυση για βέλτιστη διαδρομή μέσω του αλγορίθμου Lin-Kernighan προκύπτουν τα εξής αποτελέσματα:

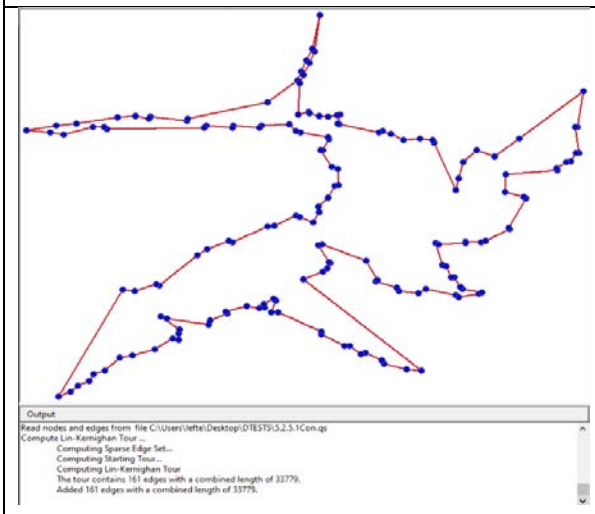
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

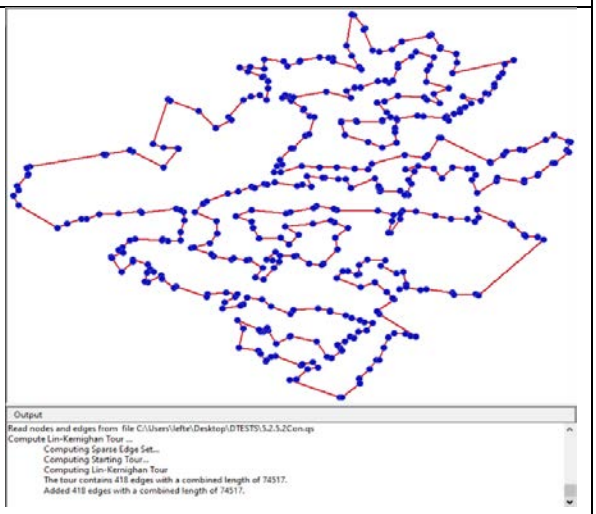
Σειρά: 3.1 / 3.2 / 3.3 / 3.4 / 3.5



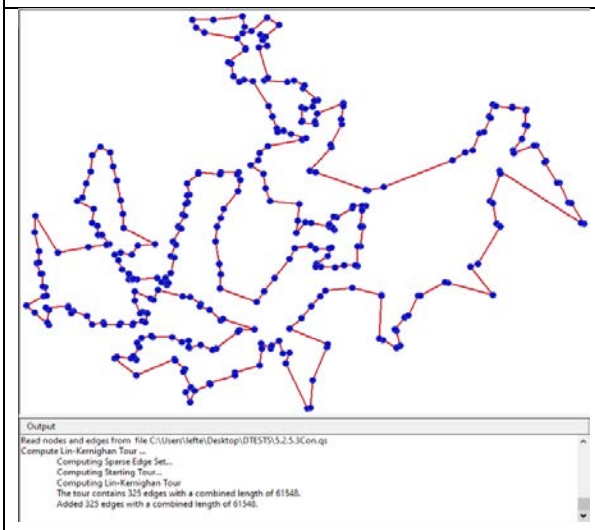
Συστάδα 3.1



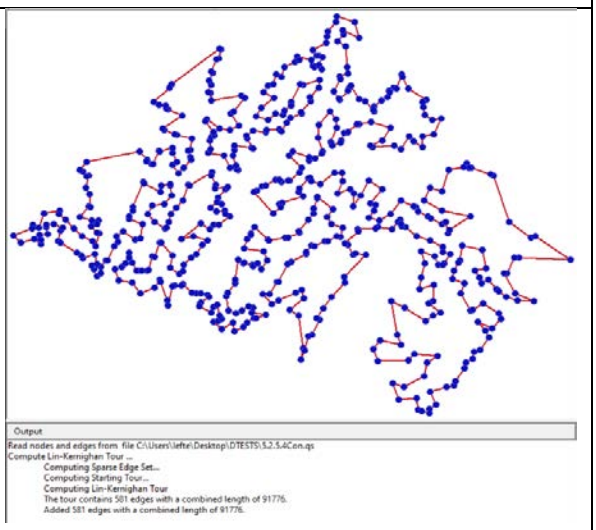
Συστάδα 3.2



Συστάδα 3.3



Συστάδα 3.4



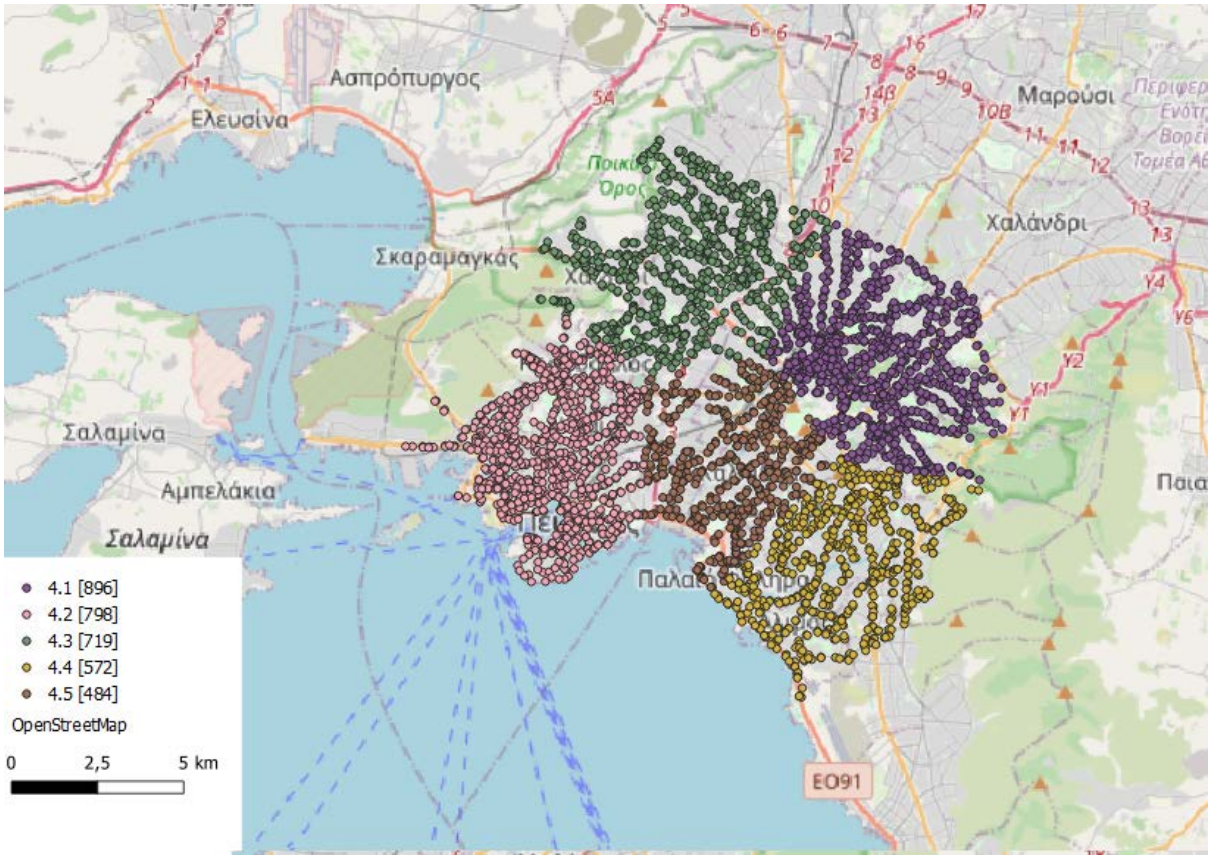
Συστάδα 3.5

Και στον παρακάτω πίνακα βλέπουμε την χρονική ανάλυση:

Πίνακας 4.4 αποτελέσματα συστάδων 3.1,3.2,3.3,3.4,3.5

Cluster Size	Minimum Distance(m)	Time(fast) in hours	Time(slow) in hours
161	33779	7.750183333	15.50036667
325	61548	14.64886667	29.29773333
418	74517	18.14421667	36.28843333
511	83255	21.00491667	42.00983333
581	91776	23.44973333	46.89946667

- **Ανάλυση τέταρτης συστάδας 3496 στάσεων:**



Σχήμα 4.10 Δεύτερο επίπεδο ανάλυσης της 4ης συστάδας

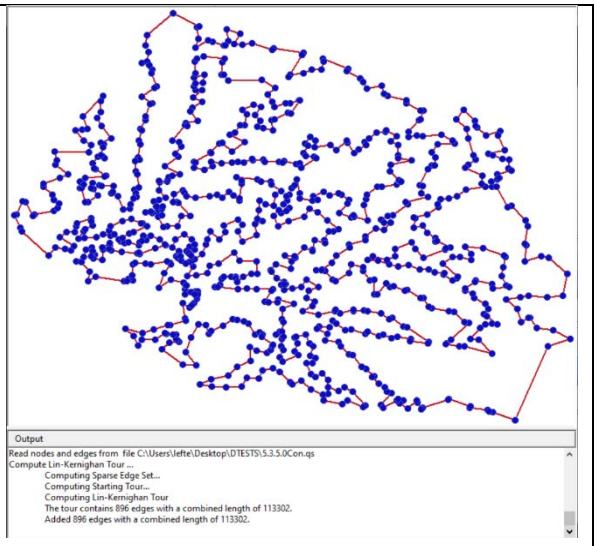
Υποδιαιρούμε την τέταρτη συστάδα μας σε υποσυστάδες μεγέθους 896, 798, 719, 572, 484 στάσεων.

Από την ανάλυση για βέλτιστη διαδρομή μέσω του αλγορίθμου Lin-Kernighan προκύπτουν τα εξής αποτελέσματα:

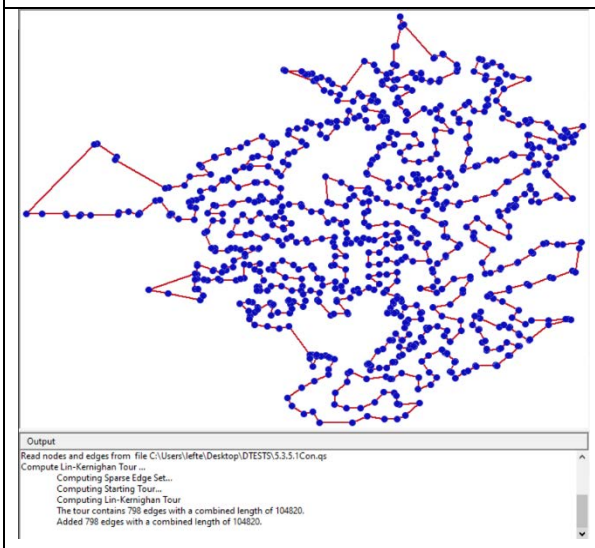
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

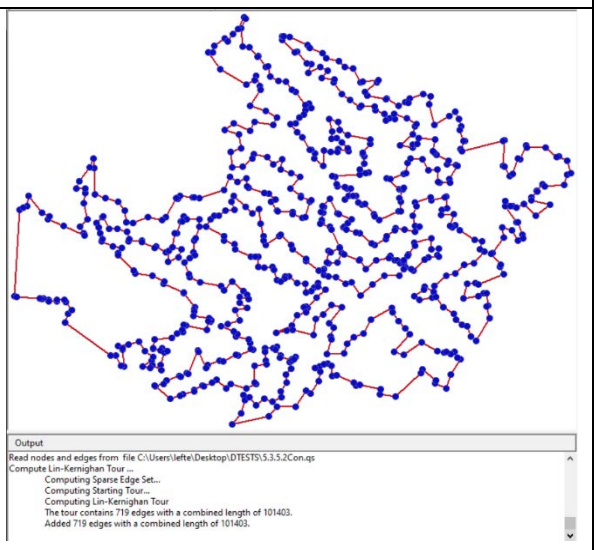
Σειρά: 4.1 / 4.2 / 4.3 / 4.4 / 4.5



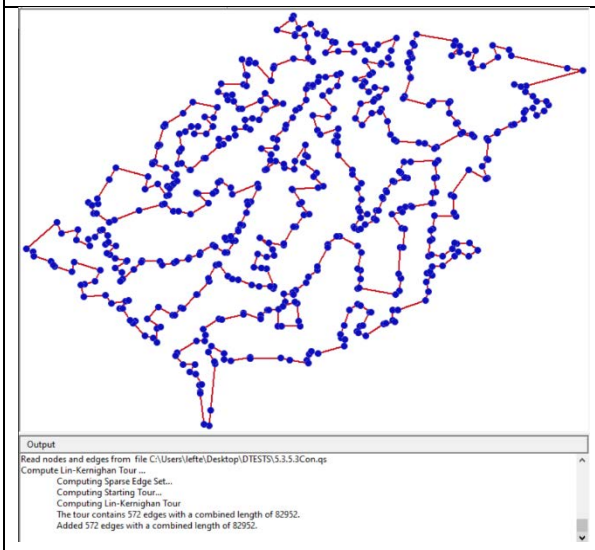
Συστάδα 4.1



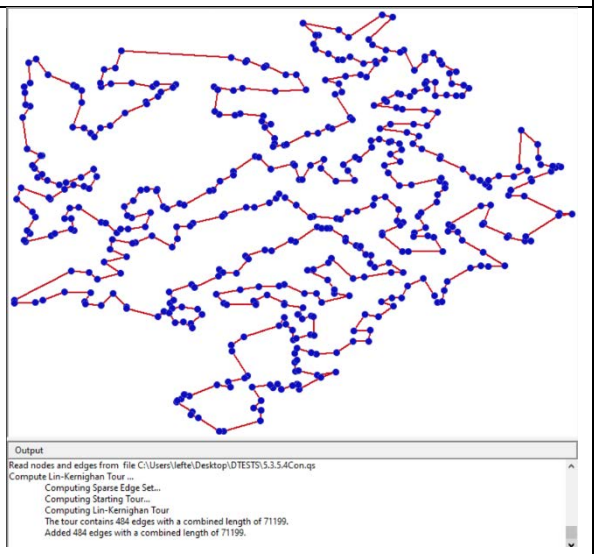
Συστάδα 4.2



Συστάδα 4.3



Συστάδα 4.4



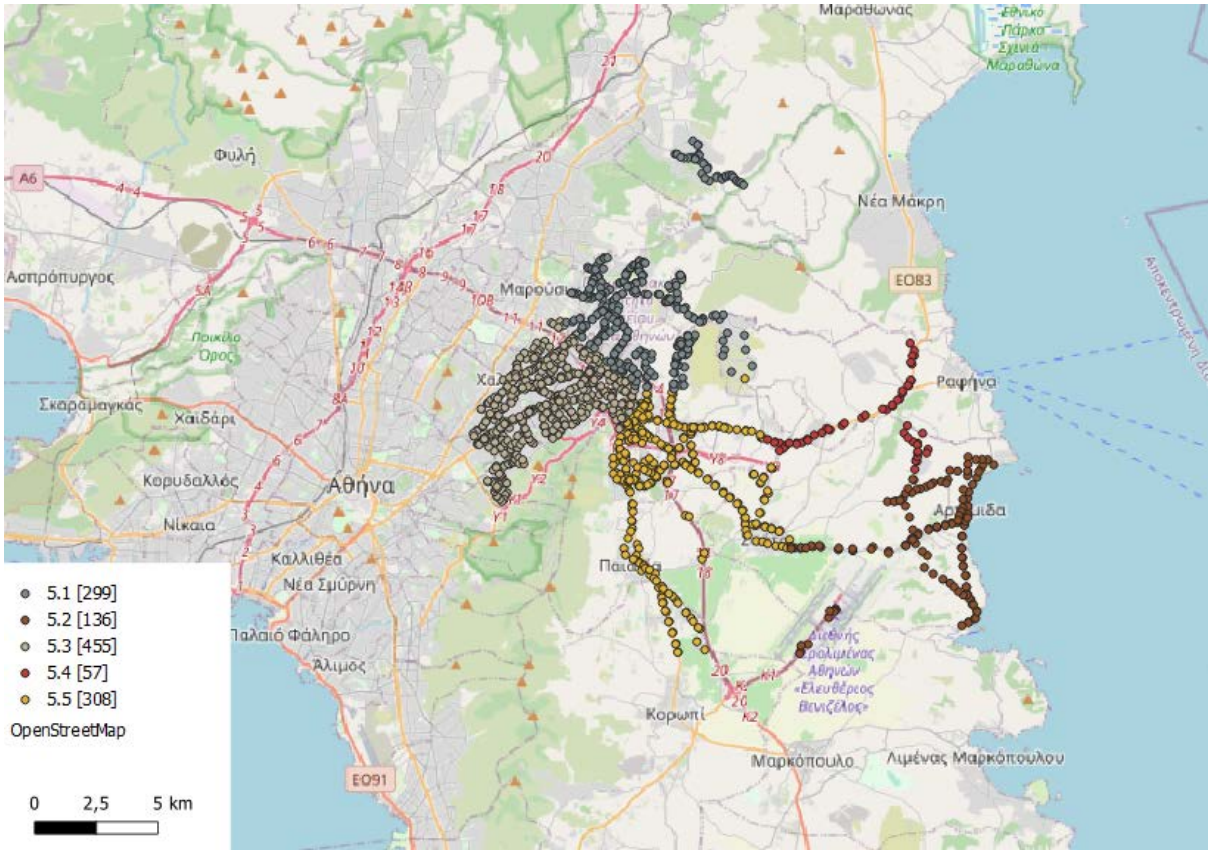
Συστάδα 4.5

Και στον παρακάτω πίνακα βλέπουμε την χρονική ανάλυση:

Πίνακας 4.5 αποτελέσματα συστάδων 4.1,4.2,4.3,4.4,4.5

Cluster Size	Minimum Distance(m)	Time(fast) in hours	Time(slow) in hours
484	71199	18.74651667	37.49303333
572	82952	21.97613333	43.95226667
719	101403	27.19378333	54.38756667
798	104820	29.023	58.046
896	113302	31.92863333	63.85726667

- **Ανάλυση πέμπτης συστάδας 3496 στάσεων:**



Σχήμα 4.11 Δεύτερο επίπεδο ανάλυσης της 5ης συστάδας

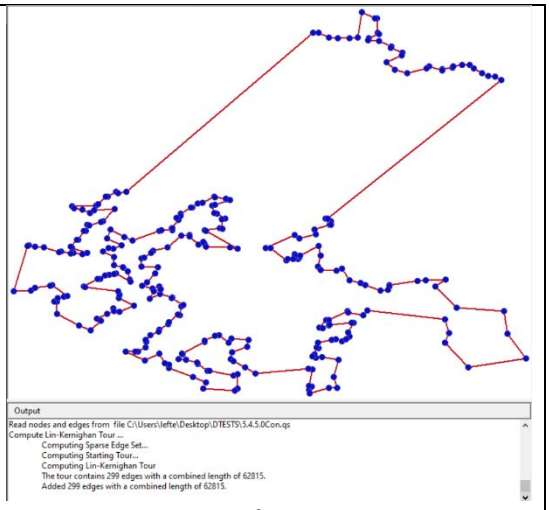
Υποδιαιρούμε την τέταρτη συστάδα μας σε υποσυστάδες μεγέθους 299, 136, 455, 57, 308 στάσεων.

Από την ανάλυση για βέλτιστη διαδρομή μέσω του αλγορίθμου Lin-Kernighan προκύπτουν τα εξής αποτελέσματα:

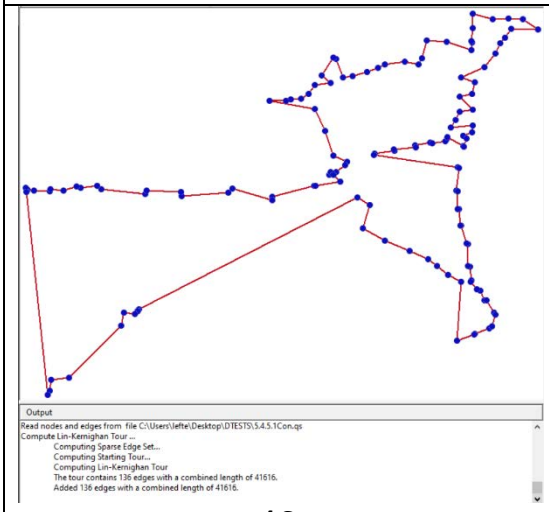
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

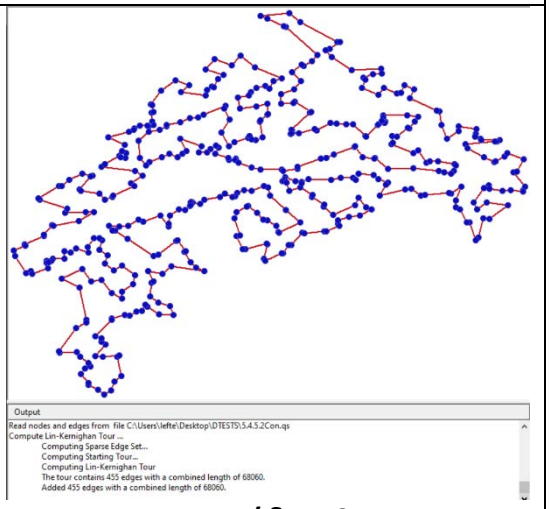
Σειρά: 5.1 / 5.2 / 5.3 / 5.4 / 5.5



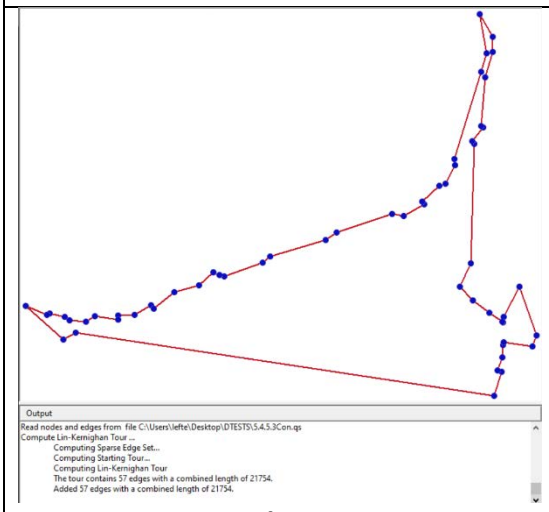
Συστάδα 5.1



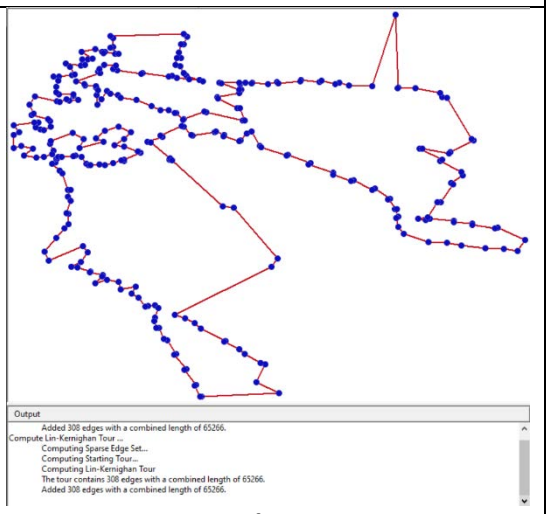
Συστάδα 5.2



Συστάδα 5.3



Συστάδα 5.4



Συστάδα 5.5

Και στον παρακάτω πίνακα βλέπουμε την χρονική ανάλυση:

Πίνακας 4.6 αποτελέσματα συστάδων 5.1,5.2,5.3,5.4,5.5

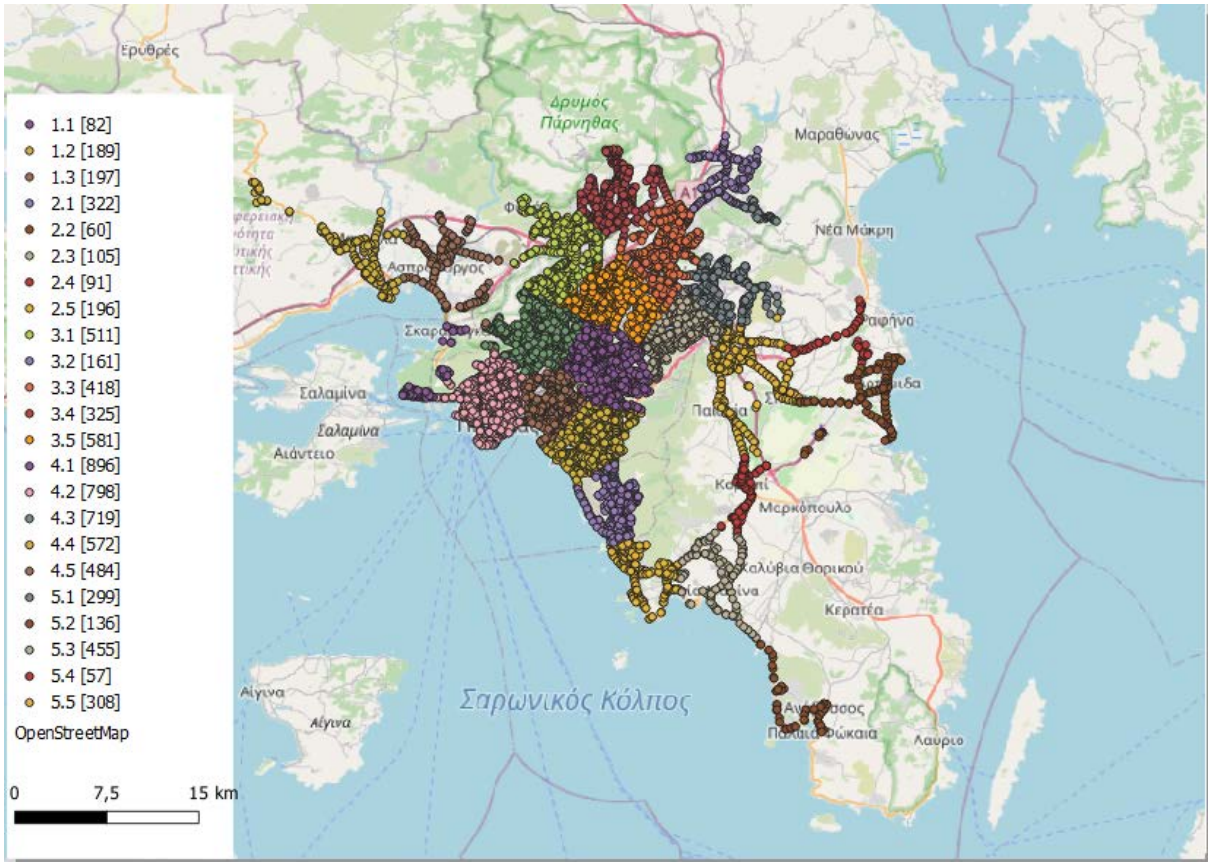
Cluster Size	Minimum Distance(m)	Time(fast) in hours	Time(slow) in hours
57	21754	4.2131	8.4262
136	41616	8.509066667	17.01813333
299	62815	14.40558333	28.81116667
308	65266	14.92323333	29.84646667
455	68060	17.79233333	35.58466667

Και συνοπτικά όλα τα αποτελέσματα του δευτέρου επιπέδου ανάλυσης τα βλέπουμε στον παρακάτω πίνακα:

Πίνακας 4.7 ολικά αποτελέσματα δεύτερου επιπέδου ανάλυσης

Αριθμός Συστάδας	Μέγεθος	Απόσταση(m)	Χρόνος(Hours) (Γρήγορο)	Χρόνος(Hours) (αργό)
1.1	82	22504	4.742266667	9.484533333
1.2	189	49803	10.62045	21.2409
1.3	197	47884	10.46593333	20.93186667
2.1	322	48417	12.62921667	25.25843333
2.2	60	23995	4.59925	9.1985
2.3	105	37487	7.37305	14.7461
2.4	91	19981	4.513816667	9.027633333
2.5	196	37403	8.877116667	17.75423333
3.1	511	83255	21.00491667	42.00983333
3.2	161	33779	7.750183333	15.50036667
3.3	418	74517	18.14421667	36.28843333
3.4	325	61548	14.64886667	29.29773333
3.5	581	91776	23.44973333	46.89946667
4.1	896	113302	31.92863333	63.85726667
4.2	798	104820	29.023	58.046
4.3	719	101403	27.19378333	54.38756667
4.4	572	82952	21.97613333	43.95226667
4.5	484	71199	18.74651667	37.49303333
5.1	299	62815	14.40558333	28.81116667
5.2	136	41616	8.509066667	17.01813333
5.3	455	68060	17.79233333	35.58466667
5.4	57	21754	4.2131	8.4262
5.5	308	65266	14.92323333	29.84646667

Και η υποδιαίρεση του δικτύου στον χάρτη μας μοιάζει ως εξής:



Σχήμα 4.12 Συστάδες δεύτερου επιπέδου ανάλυσης

4.4 Τρίτο επίπεδο ανάλυσης

Για το τρίτο επίπεδο ανάλυσης θα ακολουθήσουμε όμοια στρατηγική, δηλαδή θα υποδιαιρέσουμε τις συστάδες μας σε ακόμα μικρότερες βάσει της ώρας που χρειάζεται για να ολοκληρωθεί η επιτήρηση τους κατά την γρήγορη επίλυση.

Μέχρι στιγμής το δίκτυο μας έχει υποδιαιρεθεί σε 23 συστάδες όπως βλέπουμε και στον πίνακα 4.7 (σελ. 59). Το κριτήριο πόσο ακόμα θα υποδιαιρεθεί η κάθε μια από τις 23 συστάδες μας είναι το πόσο χρόνο απαιτεί για να ολοκληρωθεί η επιθεώρηση της. Θα διαιρέσουμε δηλαδή τον χρόνο που απαιτείται κατά την γρήγορη επίλυση με τον αριθμό 8 και έπειτα από στρογγυλοποίηση θα δούμε τι αριθμό θα επιλέξουμε στο input Clusters του αλγορίθμου k – means clustering στο QGIS. Τελικώς η στρατηγική μας της επιπλέον υποδιαίρεσης φαίνεται στον παρακάτω πίνακα:

Πίνακας 4.8 Υποδιαίρεση κάθε συστάδας για την δημιουργία του τρίτου επιπέδου

Αριθμός Συστάδας	Χρόνος(Γρηγορο)	Υποδιαίρεση
1.1	4.742266667	1
1.2	10.62045	1
1.3	10.46593333	1
2.1	12.62921667	2
2.2	4.59925	1
2.3	7.37305	1
2.4	4.513816667	1
2.5	8.877116667	1
3.1	21.00491667	3
3.2	7.750183333	1
3.3	18.14421667	2
3.4	14.64886667	2
3.5	23.44973333	3
4.1	31.92863333	4
4.2	29.023	4
4.3	27.19378333	3
4.4	21.97613333	3
4.5	18.74651667	2
5.1	14.40558333	2
5.2	8.509066667	1
5.3	17.79233333	2
5.4	4.2131	1
5.5	14.92323333	2
Συνολικα:		
23		44

Οπότε βλέπουμε ότι μετά την ολοκλήρωση του τρίτου επιπέδου ανάλυσης θα έχουν προκύψει συνολικά 44 συστάδες.

Η διαδικασία είναι όμοια όπως περιγραφικέ προηγουμένως και χρησιμοποιούμαι τις ίδιες λειτουργίες που δείξαμε στο τρίτο κεφάλαιο.

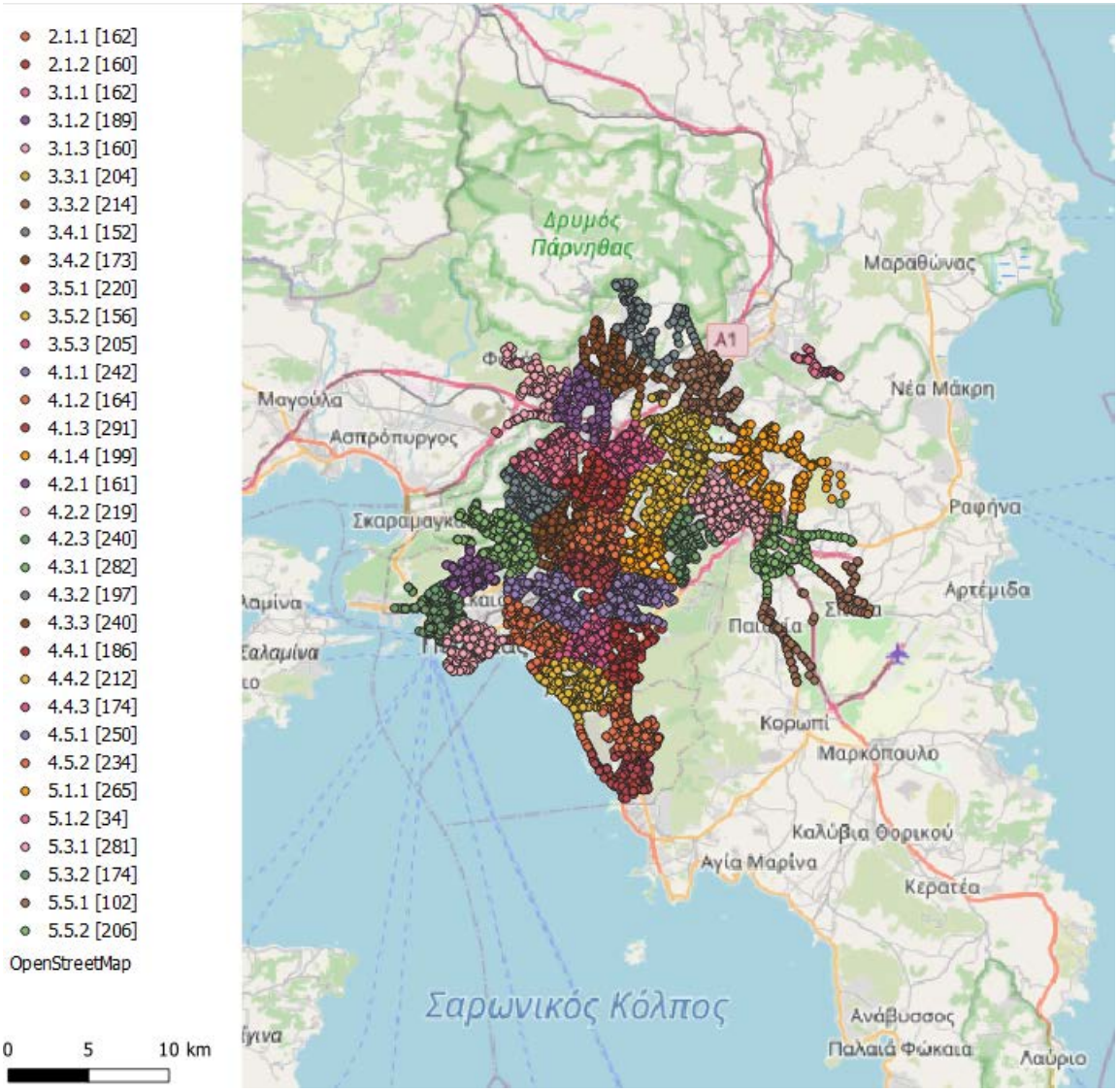
Τελικώς με την χρήση του εργαλείου k – means clustering στο QGIS θα αποκτήσουμε τις εξής καινούργιες συστάδες για τις οποίες θα πρέπει να υπολογίσουμε την βέλτιστη διαδρομή και τον χρόνο ολοκλήρωσης της επιθεώρησης τους.

Εδώ βλέπουμε τον πίνακα με το σύνολο των συστάδων που δημιουργήσαμε και τον αριθμό των στάσεων που περιέχουν:

Πίνακας 4.9 Συστάδες τρίτου επιπέδου

Συστάδα	Μέγεθος
2.1.1	162
2.1.2	160
3.1.1	162
3.1.2	189
3.1.3	160
3.3.1	204
3.3.2	214
3.4.1	152
3.4.2	173
3.5.1	220
3.5.2	156
3.5.3	205
4.1.1	242
4.1.2	164
4.1.3	291
4.1.4	199
4.2.1	161
4.2.2	219
4.2.3	240
4.2.4	178
4.3.1	282
4.3.2	197
4.3.3	240
4.4.1	186
4.4.2	212
4.4.3	174
4.5.1	250
4.5.2	234
5.1.1	265
5.1.2	34
5.3.1	281
5.3.2	174
5.5.1	102
5.5.2	206
Συνολικά	
34	

Και εδώ βλέπουμε την διάταξη τους στον χάρτη:



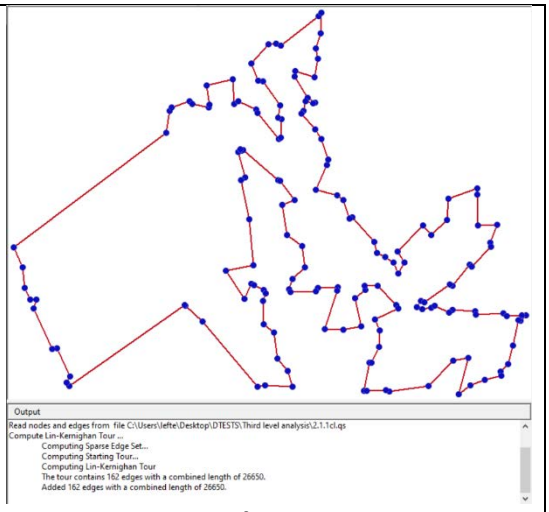
<u>Σχήμα 4.13 Συστάδες τρίτου επιπέδου ανάλυσης</u>

Στην συνέχεια θα παρουσιάσουμε τις λύσεις που έδωσε το Concorde TSP Solver για βέλτιστη δρομολόγηση του οχήματος του επιθεωρητή στην κάθε μια.

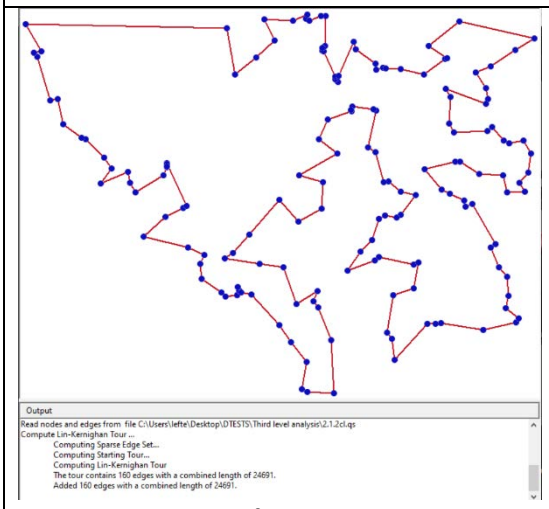
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

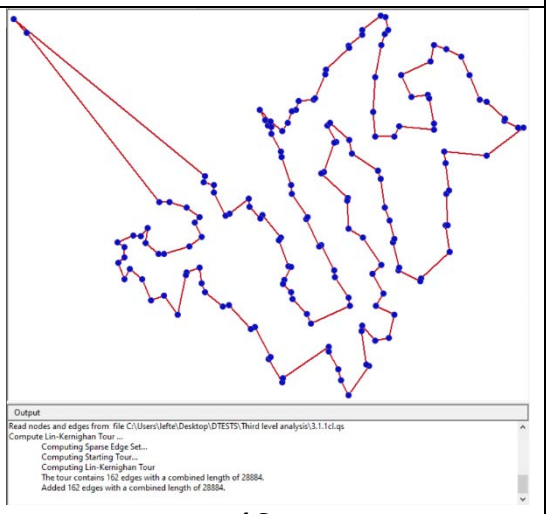
Σειρά: 2.1.1 / 2.1.2 / 3.1.1 / 3.1.2 / 3.1.3



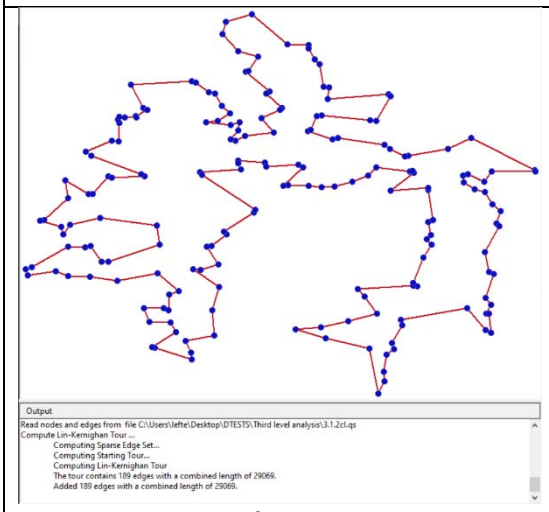
Συστάδα 2.1.1



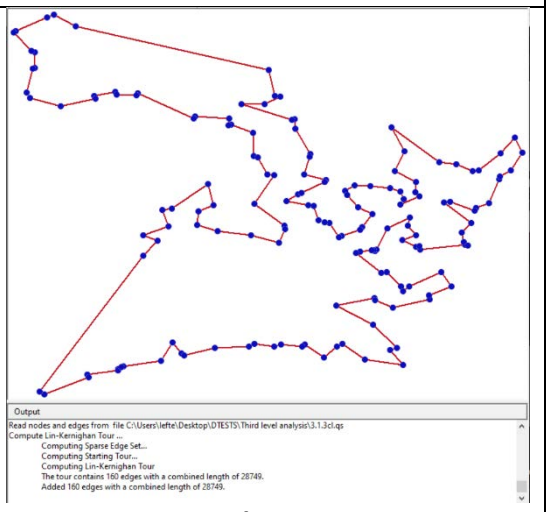
Συστάδα 2.1.2



Συστάδα 3.1.1



Συστάδα 3.1.2

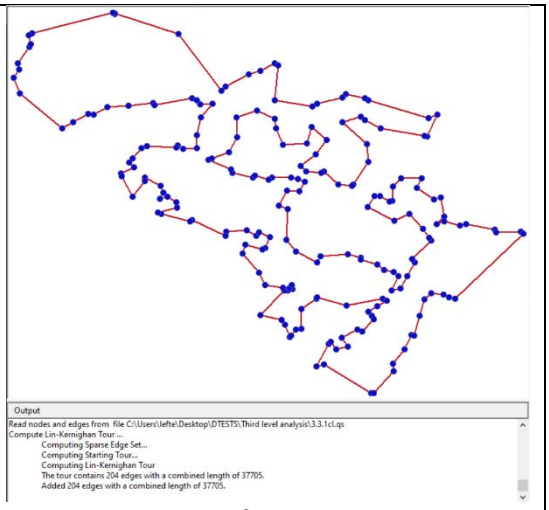


Συστάδα 3.1.3

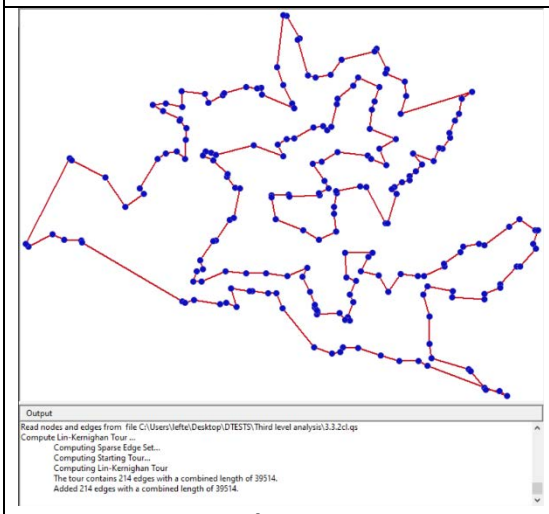
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

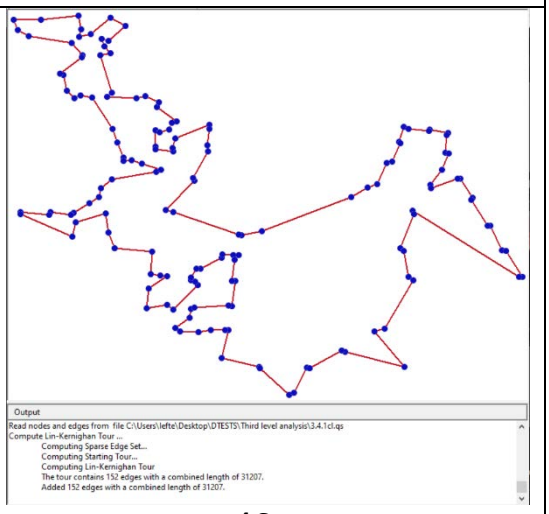
Σειρά: 3.3.1/3.3.2/3.4.1/3.4.2/3.5.1



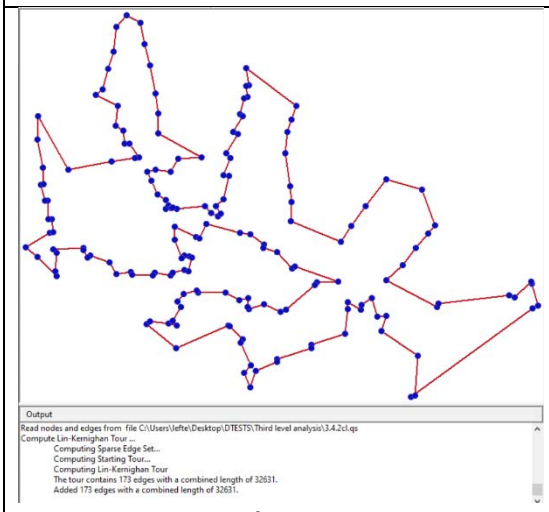
Συστάδα 3.3.1



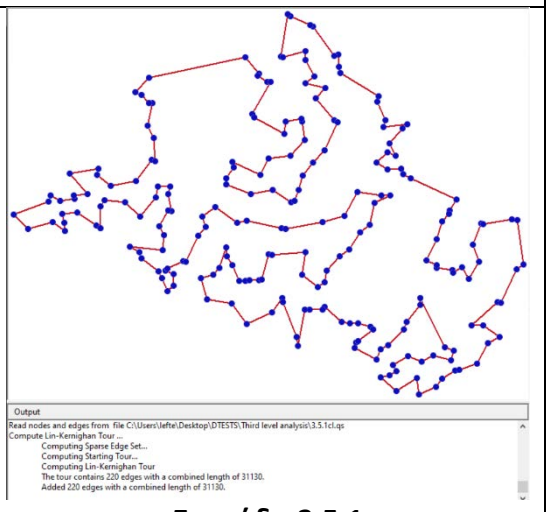
Συστάδα 3.3.2



Συστάδα 3.4.1



Συστάδα 3.4.2

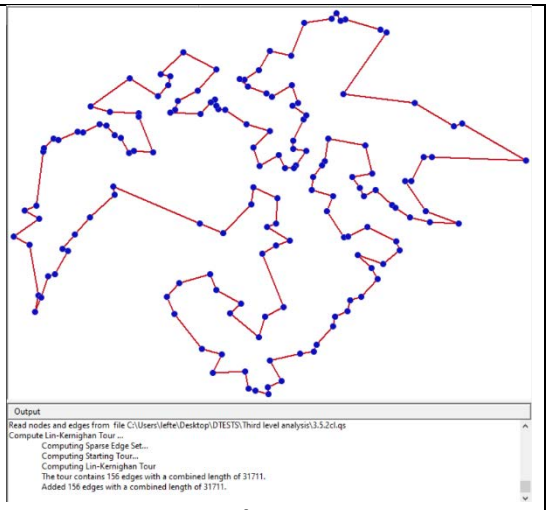


Συστάδα 3.5.1

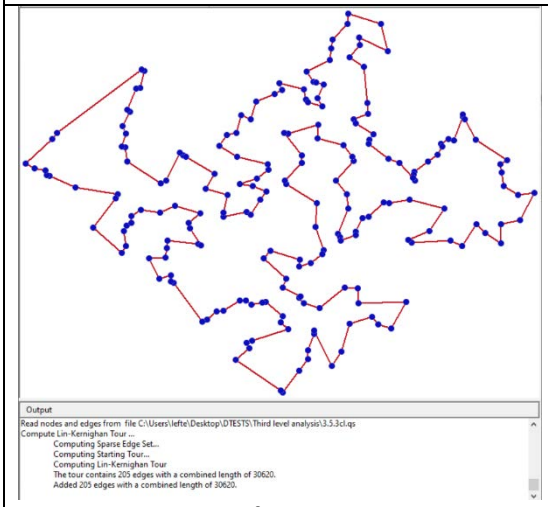
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

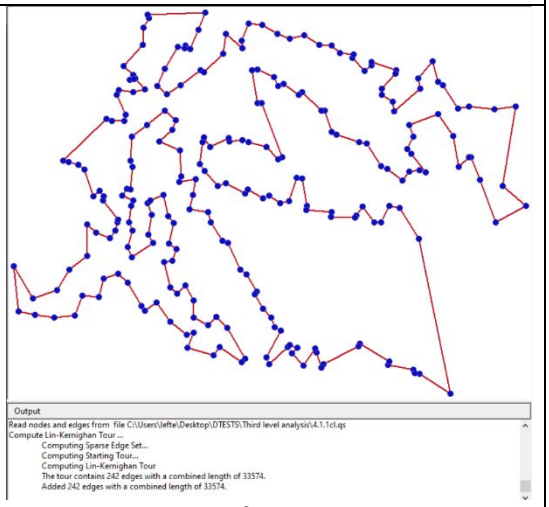
Σειρά: 3.5.2/3.5.3/4.1.1/4.1.2/4.1.3



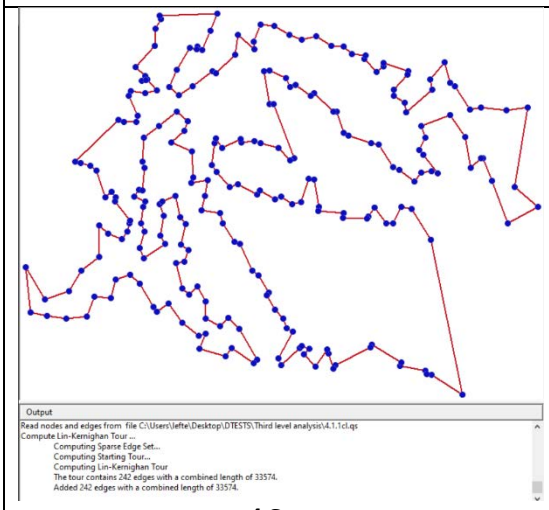
Συστάδα 3.5.2



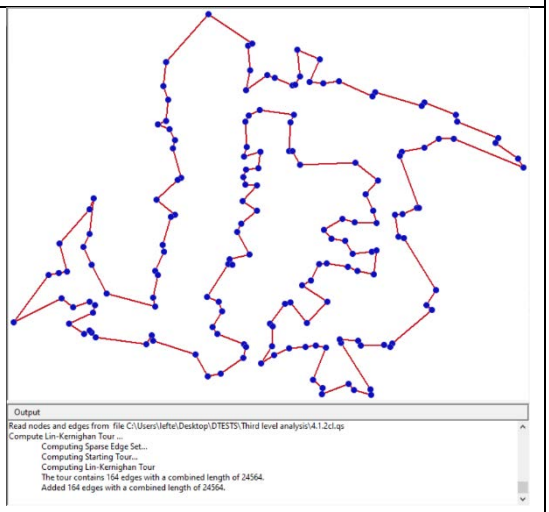
Συστάδα 3.5.3



Συστάδα 4.1.1



Συστάδα 4.1.2

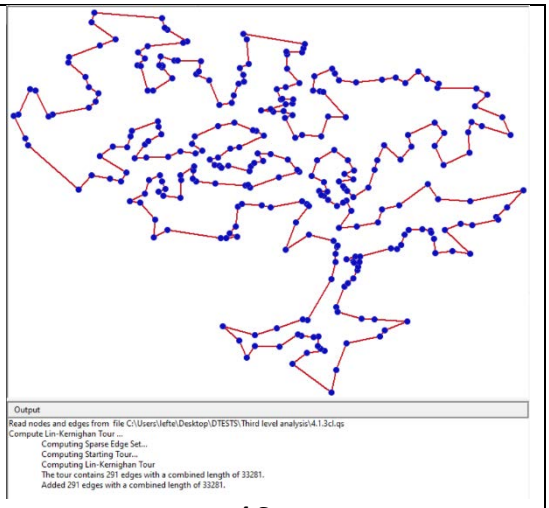


Συστάδα 4.1.3

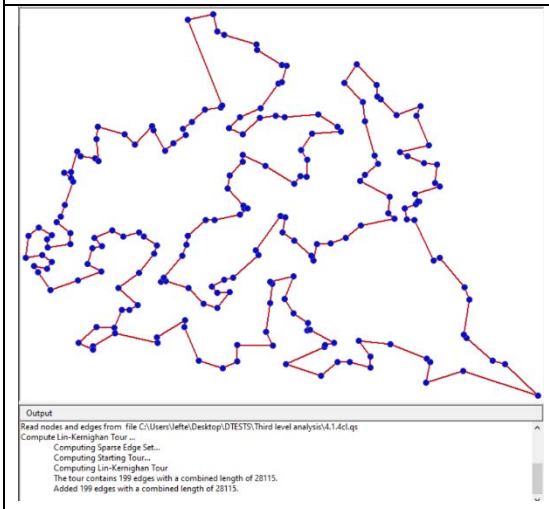
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

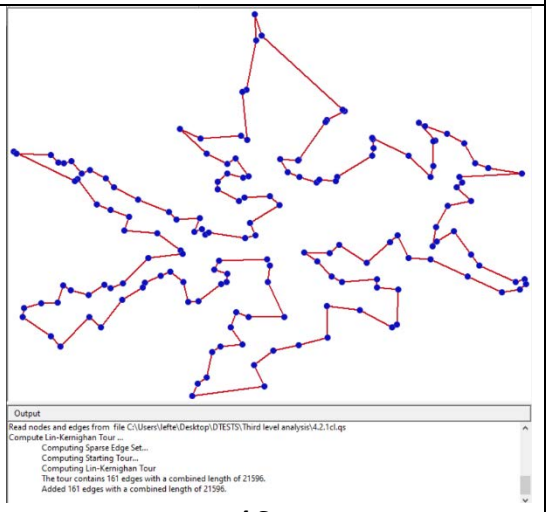
Σειρά: 4.1.4/4.2.1/4.2.2/4.2.3/4.2.4



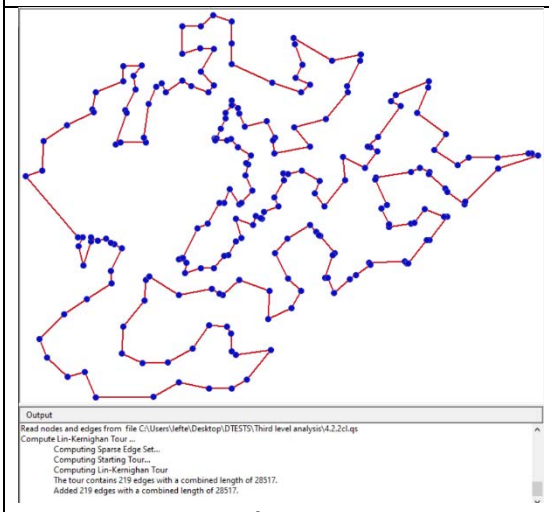
Συστάδα 4.1.4



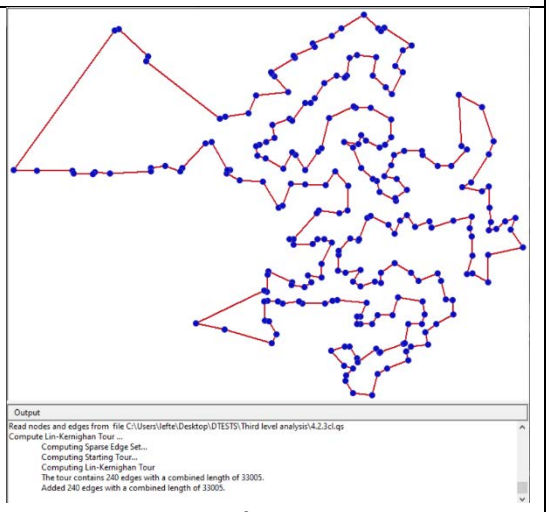
Συστάδα 4.2.1



Συστάδα 4.2.2



Συστάδα 4.2.3

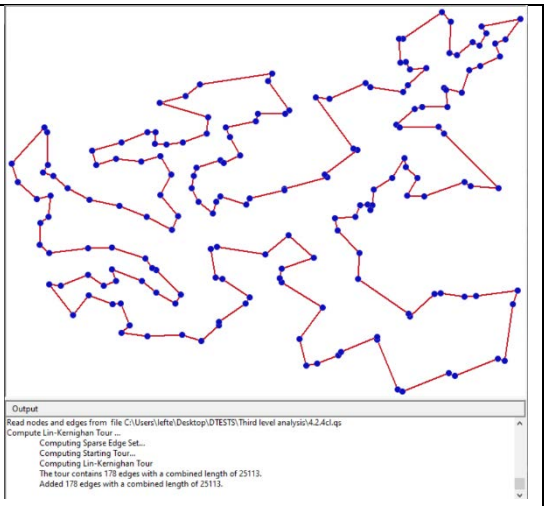


Συστάδα 4.2.4

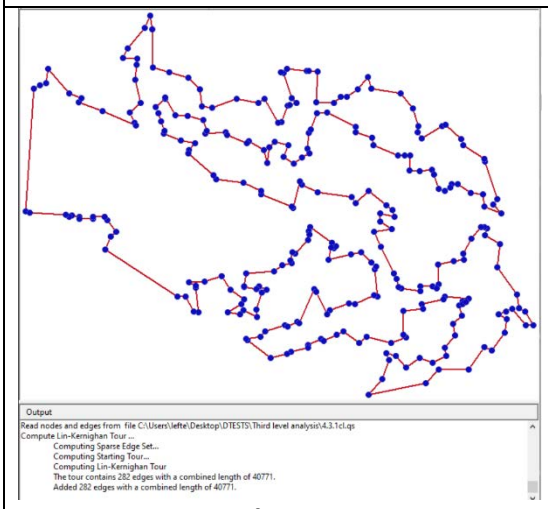
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

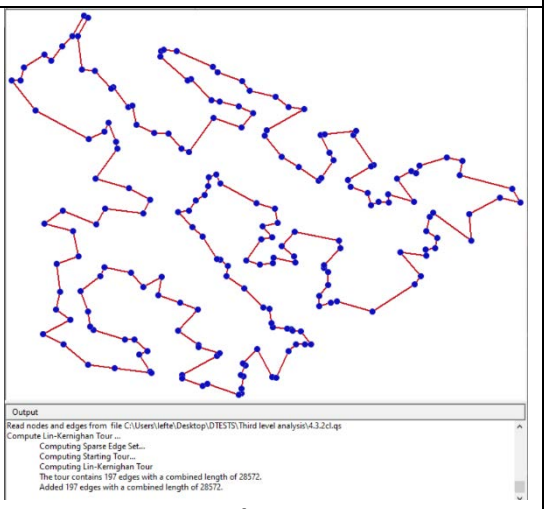
Σειρά: 4.3.1/4.3.2/4.3.3/4.4.1/4.4.2



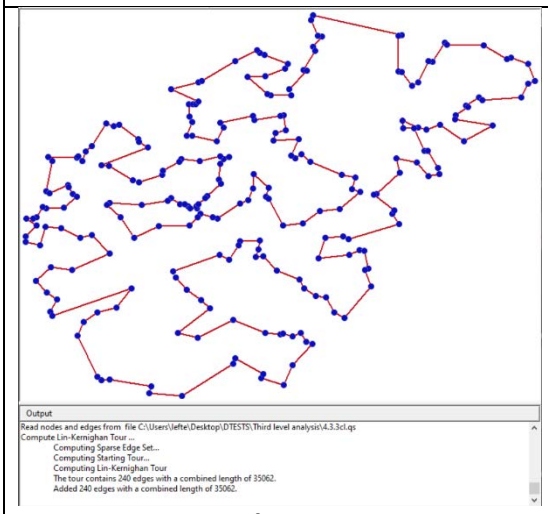
Συστάδα 4.3.1



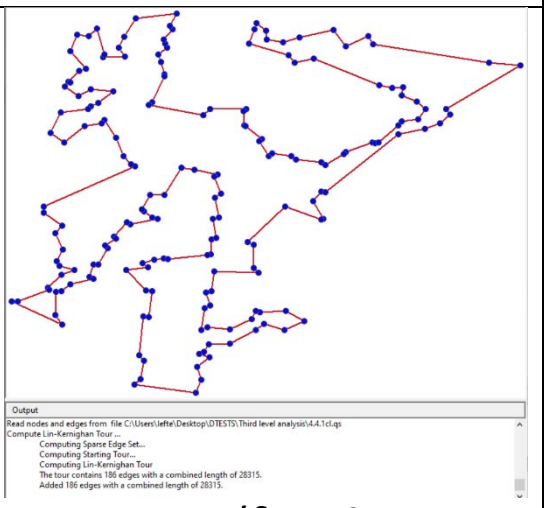
Συστάδα 4.3.2



Συστάδα 4.3.3



Συστάδα 4.4.1

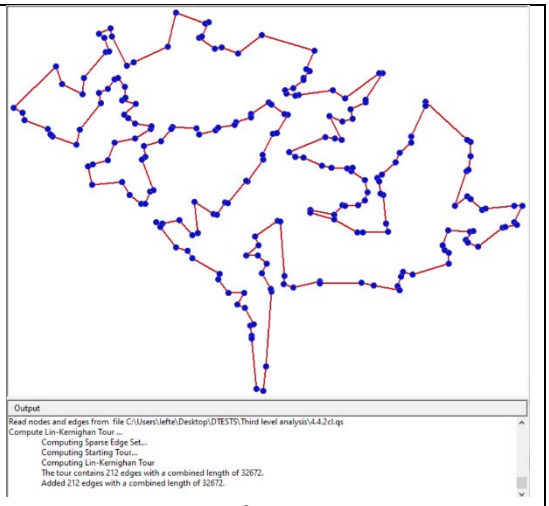


Συστάδα 4.4.2

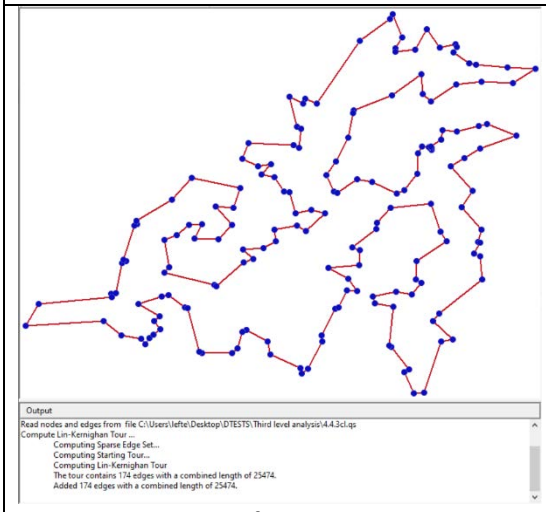
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

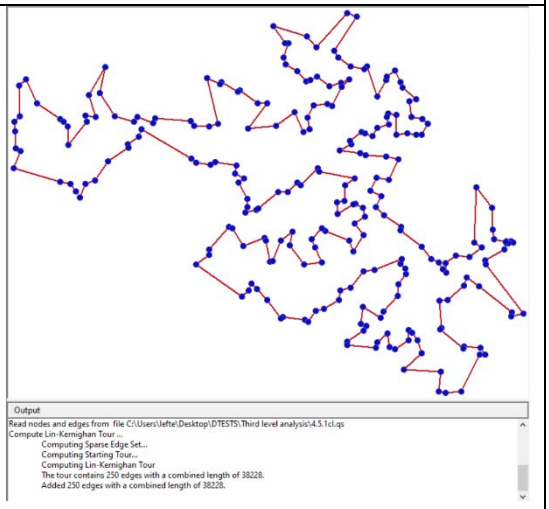
Σειρά: 4.4.3/4.5.1/4.5.2/5.1.1/5.1.2



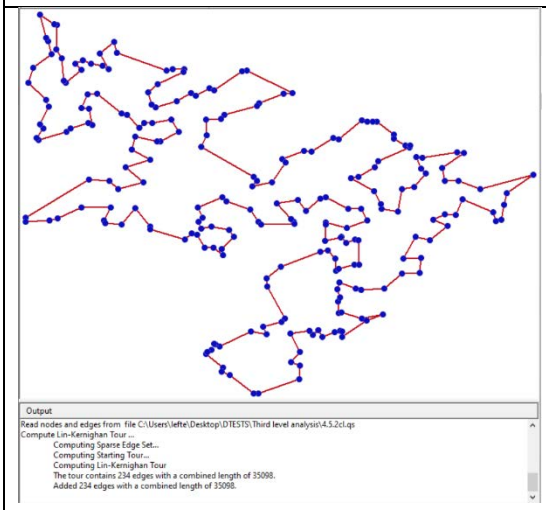
Συστάδα 4.4.3



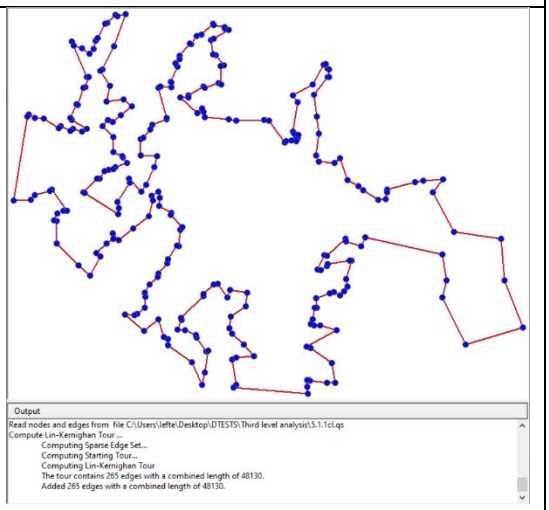
Συστάδα 4.5.1



Συστάδα 4.5.2



Συστάδα 5.1.1

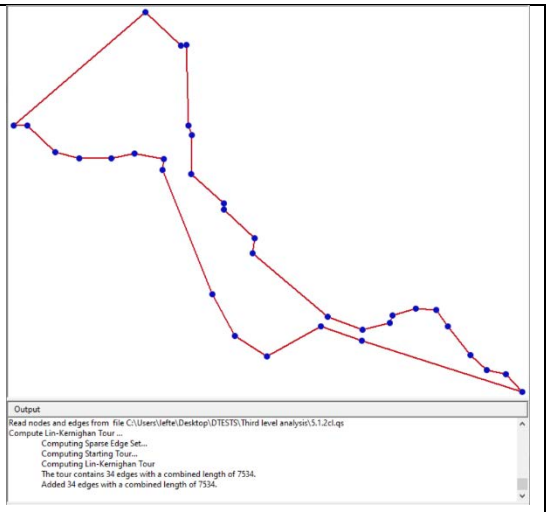


Συστάδα 5.1.2

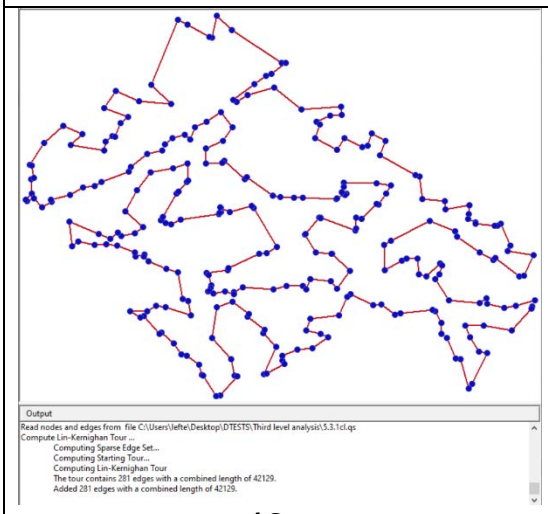
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

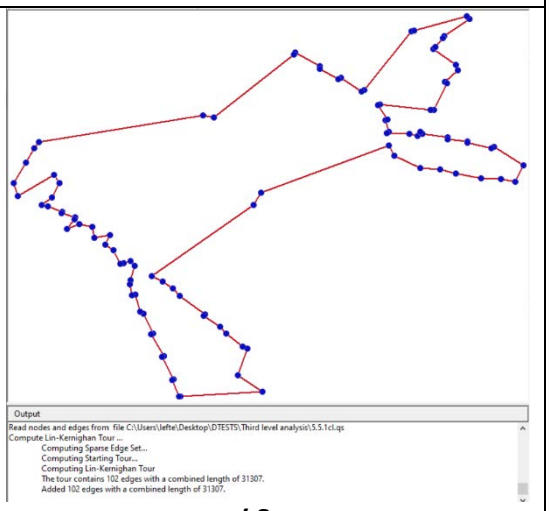
Σειρά: 5.3.1/5.3.2/5.5.1/5.5.2



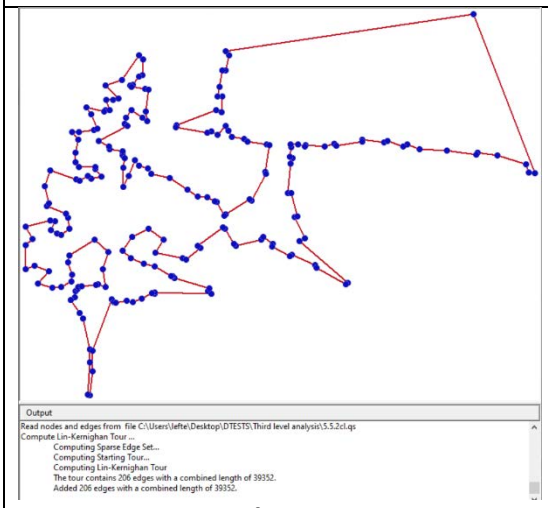
Συστάδα 5.3.1



Συστάδα 5.3.2



Συστάδα 5.5.1



Συστάδα 5.5.2

Και αν εισάγουμε σε ένα υπολογιστικό φύλλο excel τα στοιχεία μας και εφαρμόσουμε τις υποθέσεις ταχύτητας και διάρκειας επιθεώρησης στάσης για κάθε περίπτωση έχουμε ολικά τα εξής αποτελέσματα:

Πίνακας 4.8 ολικά αποτελέσματα τρίτου επιπέδου ανάλυσης

Συστάδα	Στάσεις	Απόσταση	Χρόνος(Γρήγορο)	Χρόνος(Αργό)
1.1	82	22504	4.742266667	9.484533333
1.2	189	49803	10.62045	21.2409
1.3	197	47884	10.46593333	20.93186667
2.1.1	162	26650	6.6975	13.395
2.1.2	160	24691	6.370316667	12.74063333
2.2	60	23995	4.59925	9.1985
2.3	105	37487	7.37305	14.7461
2.4	91	19981	4.513816667	9.027633333
2.5	196	37403	8.877116667	17.75423333
3.1.1	162	28884	7.0326	14.0652
3.1.2	189	29069	7.51035	15.0207
3.1.3	160	28749	6.979016667	13.95803333
3.2	161	33779	7.750183333	15.50036667
3.3.1	204	37705	9.05575	18.1115
3.3.2	214	39514	9.493766667	18.98753333
3.4.1	152	31207	7.214383333	14.42876667
3.4.2	173	32631	7.777983333	15.55596667
3.5.1	220	31130	8.336166667	16.67233333
3.5.2	156	31711	7.35665	14.7133
3.5.3	205	30620	8.009666667	16.01933333
4.1.1	242	33574	9.069433333	18.13886667
4.1.2	164	24564	6.417933333	12.83586667
4.1.3	291	33281	9.84215	19.6843
4.1.4	199	28115	7.533916667	15.06783333
4.2.1	161	21596	5.922733333	11.84546667
4.2.2	219	28517	7.92755	15.8551
4.2.3	240	33005	8.95075	17.9015
4.2.4	178	25113	6.733616667	13.46723333
4.3.1	282	40771	10.81565	21.6313
4.3.2	197	28572	7.569133333	15.13826667
4.3.3	240	35062	9.2593	18.5186
4.4.1	186	28315	7.34725	14.6945
4.4.2	212	32672	8.434133333	16.86826667
4.4.3	174	25474	6.7211	13.4422
4.5.1	250	38228	9.900866667	19.80173333
4.5.2	234	35098	9.1647	18.3294
5.1.1	265	48130	11.63616667	23.27233333
5.1.2	34	7534	1.696766667	3.393533333
5.2	136	41616	8.509066667	17.01813333
5.3.1	281	42129	11.00268333	22.00536667

5.3.2	174	27245	6.98675	13.9735
5.4	57	21754	4.2131	8.4262
5.5.1	102	31037	6.35555	12.7111
5.5.2	206	39532	9.363133333	18.72626667

Και εδώ τελειώνει η ανάλυση μας όσων αφορά την γρήγορη επίλυση. Χρησιμοποιήσαμε τον k – means αλγόριθμο με την διαδικασία της εκ νέου συνεχής συσταδιοποίησης και αυτά είναι τα αποτελέσματα που είναι όσων το δυνατόν πιο κοντά στις 8 ώρες με αυτή την στρατηγική.

Θα προχωρήσουμε σε επιπλέον ανάλυση για την περίπτωση της αργής επίλυσης καθώς παρατηρούμε πλήθος συστάδων που απαιτούν πάνω από 8 ώρες για να επιθεωρηθούν.

4.5 Τέταρτο επίπεδο ανάλυσης – Επιπλέον συσταδιοποίηση για τις ανάγκες της αργής επίλυσης

Από την έως τώρα ανάλυση έχουμε 44 συστάδες. Στον παρακάτω πίνακα βλέπουμε τις συστάδες μας και πόση ώρα χρειάζονται κατά την αργή επίλυση για να επιθεωρηθούν. Έπειτα όμοια και με την στρατηγική που έχουμε ακολουθήσει έως τώρα βρίσκουμε τον αριθμό τις επιπλέον υποδιαίρεσης διαιρώντας τον χρόνο που χρειάζονται με τον αριθμό 8 και στρογγυλοποιώντας. Στον παρακάτω πίνακα βλέπουμε τις 44 συστάδες και τον αριθμό της επιπλέον υποδιαίρεσης που θα πραγματοποιήσουμε:

Πίνακας 4.9 υποδιαίρεση συστάδων για το τέταρτο επίπεδο ανάλυσης

Συστάδα	Στάσεις	Χρόνος(Αργό)	Υποδιαίρεση
1.1	82	9.484533333	1
1.2	189	21.2409	3
1.3	197	20.93186667	3
2.1.1	162	13.395	2
2.1.2	160	12.74063333	2
2.2	60	9.1985	1
2.3	105	14.7461	2
2.4	91	9.027633333	1
2.5	196	17.75423333	2
3.1.1	162	14.0652	2
3.1.2	189	15.0207	2
3.1.3	160	13.95803333	2
3.2	161	15.50036667	2
3.3.1	204	18.1115	2
3.3.2	214	18.98753333	2
3.4.1	152	14.42876667	2

3.4.2	173	15.55596667	2
3.5.1	220	16.67233333	2
3.5.2	156	14.7133	2
3.5.3	205	16.01933333	2
4.1.1	242	18.13886667	2
4.1.2	164	12.83586667	2
4.1.3	291	19.6843	2
4.1.4	199	15.06783333	2
4.2.1	161	11.84546667	2
4.2.2	219	15.8551	2
4.2.3	240	17.9015	2
4.2.4	178	13.46723333	2
4.3.1	282	21.6313	3
4.3.2	197	15.13826667	2
4.3.3	240	18.5186	2
4.4.1	186	14.6945	2
4.4.2	212	16.86826667	2
4.4.3	174	13.4422	2
4.5.1	250	19.80173333	2
4.5.2	234	18.3294	2
5.1.1	265	23.27233333	3
5.1.2	34	3.393533333	1
5.2	136	17.01813333	2
5.3.1	281	22.00536667	3
5.3.2	174	13.9735	2
5.4	57	8.4262	1
5.5.1	102	12.7111	2
5.5.2	206	18.72626667	2

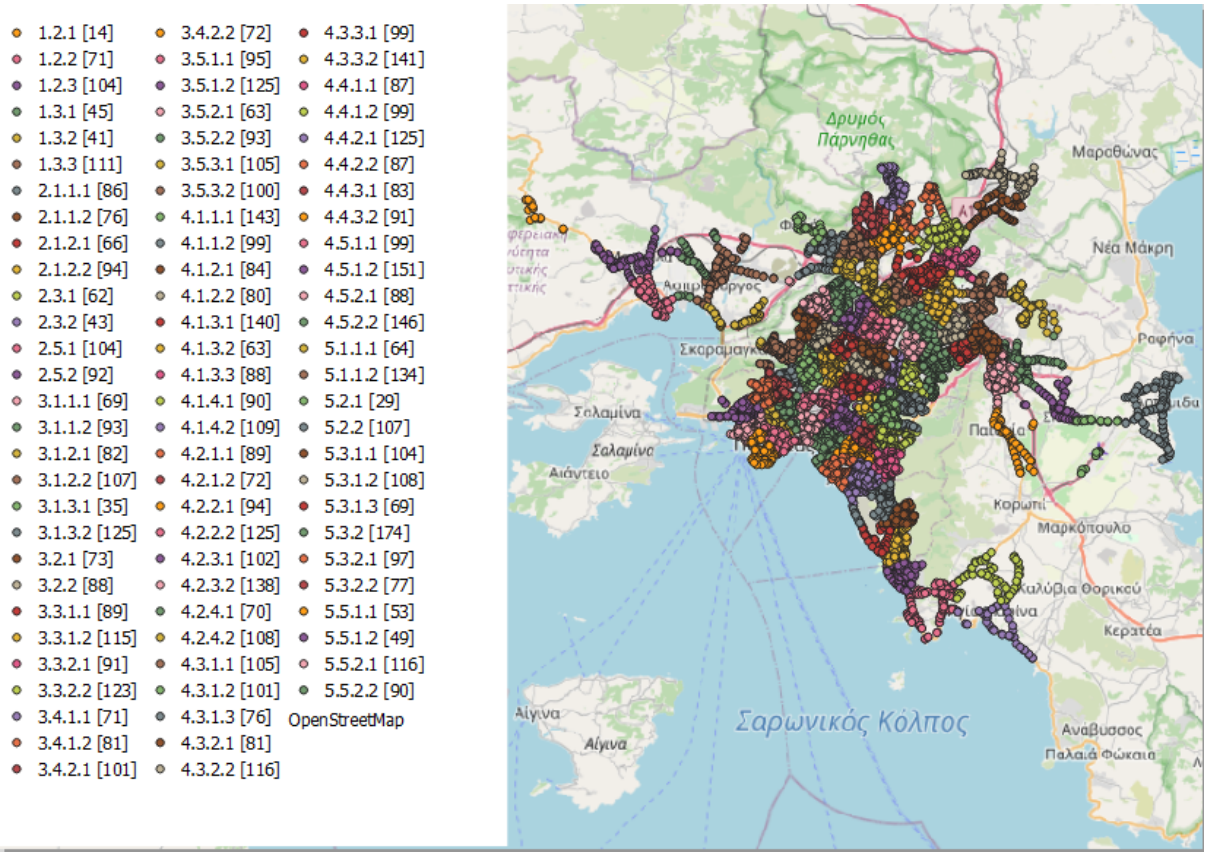
Οπότε τελικά από αυτή την διαδικασία θα πάρουμε 84 καινούργιες συστάδες τις οποίες θα πρέπει να βρούμε την βέλτιστη διαδρομή και να υπολογίσουμε τον χρόνο που απαιτούν. Στον παρακάτω πίνακα φαίνεται η αρίθμηση των 84 αυτών συστάδων που δημιουργήσαμε και αριθμός των στάσεων που περιέχουν:

Πίνακας 4.10 Συστάδες και αριθμός στάσεων τετάρτου επιπέδου ανάλυσης

Συστάδα	Στάσεις	Συστάδα	Στάσεις
1.2.1	**14**	**4.1.3.3**	**88**
1.2.2	**71**	**4.1.4.1**	**90**
1.2.3	**104**	**4.1.4.2**	**109**
1.3.1	**45**	**4.2.1.1**	**89**
1.3.2	**41**	**4.2.1.2**	**72**
1.3.3	**111**	**4.2.2.1**	**94**
2.1.1.1	**86**	**4.2.2.2**	**125**

2.1.1.2	76	4.2.3.1	102
2.1.2.1	66	4.2.3.2	138
2.1.2.2	94	4.2.4.1	70
2.3.1	62	4.2.4.2	108
2.3.2	43	4.3.1.1	105
2.5.1	104	4.3.1.2	101
2.5.2	92	4.3.1.3	76
3.1.1.1	69	4.3.2.1	81
3.1.1.2	93	4.3.2.2	116
3.1.2.1	82	4.3.3.1	99
3.1.2.2	107	4.3.3.2	141
3.1.3.1	35	4.4.1.1	87
3.1.3.2	125	4.4.1.2	99
3.2.1	73	4.4.2.1	125
3.2.2	88	4.4.2.2	87
3.3.1.1	89	4.4.3.1	83
3.3.1.2	115	4.4.3.2	91
3.3.2.1	91	4.5.1.1	99
3.3.2.2	123	4.5.1.2	151
3.4.1.1	71	4.5.2.1	88
3.4.1.2	81	4.5.2.2	146
3.4.2.1	101	5.1.1.1	64
3.4.2.2	72	5.1.1.2	134
3.5.1.1	95	5.1.1.3	67
3.5.1.2	125	5.2.1	29
3.5.2.1	63	5.2.2	107
3.5.2.2	93	5.3.1.1	104
3.5.3.1	105	5.3.1.2	108
3.5.3.2	100	5.3.1.3	69
4.1.1.1	143	5.3.2.1	97
4.1.1.2	99	5.3.2.2	77
4.1.2.1	84	5.5.1.1	53
4.1.2.2	80	5.5.1.2	49
4.1.3.1	140	5.5.2.1	116
4.1.3.2	63	5.5.2.2	90

Και στον παρακάτω χάρτη παρατηρούμε την διάταξη τους στον γεωγραφικό χώρο:



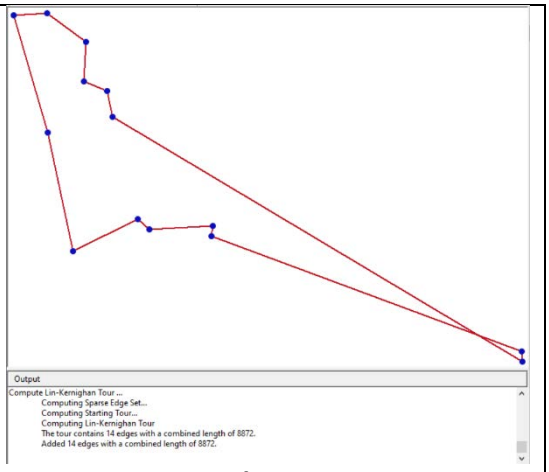
Σχήμα 4.14 Συστάδες τετάρτου επιπέδου ανάλυσης

Και στους παρακάτω πίνακες παρουσιάζεται η ανάλυση κάθε μιας συστάδας που προέκυψε από την εισαγωγή των συντεταγμένων των σημείων της στο Concorde TSP Solver:

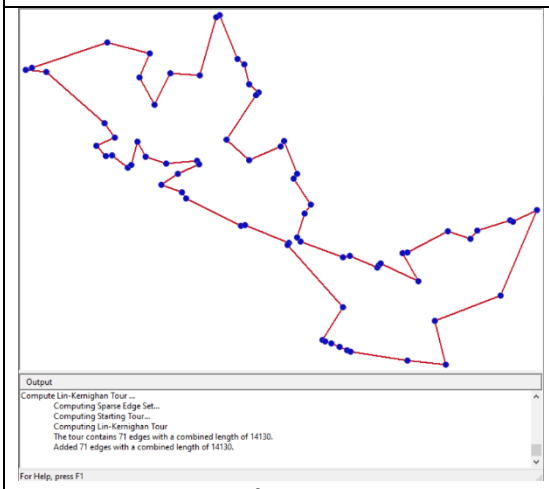
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

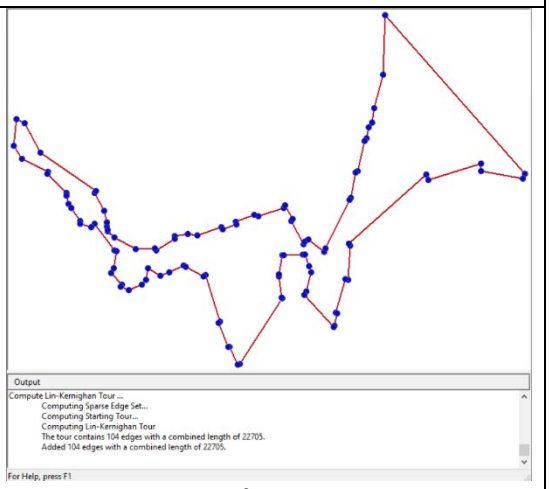
Σειρά: 1.2.1/1.2.2/1.2.3/1.3.1/1.3.2



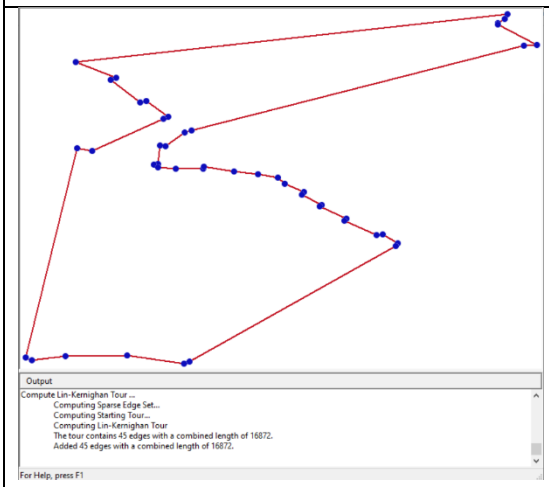
Συστάδα 1.2.1



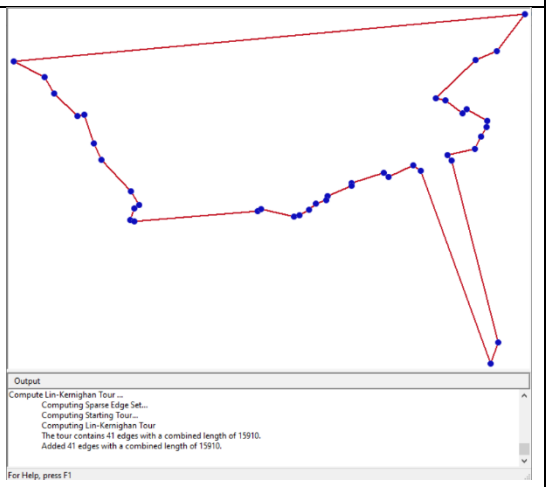
Συστάδα 1.2.2



Συστάδα 1.2.3



Συστάδα 1.3.1

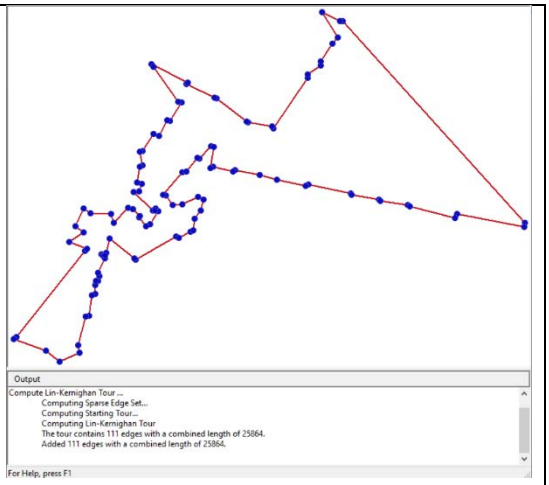


Συστάδα 1.3.2

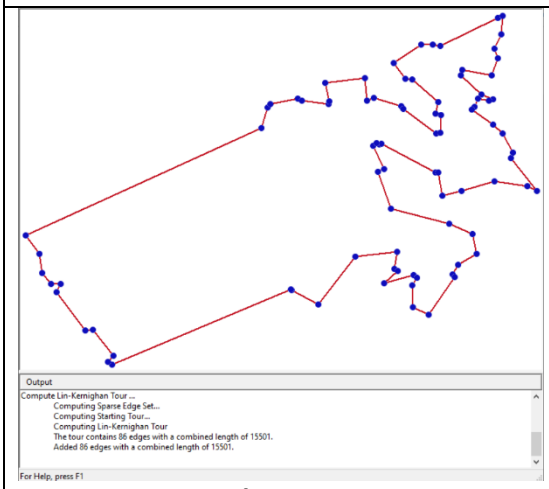
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

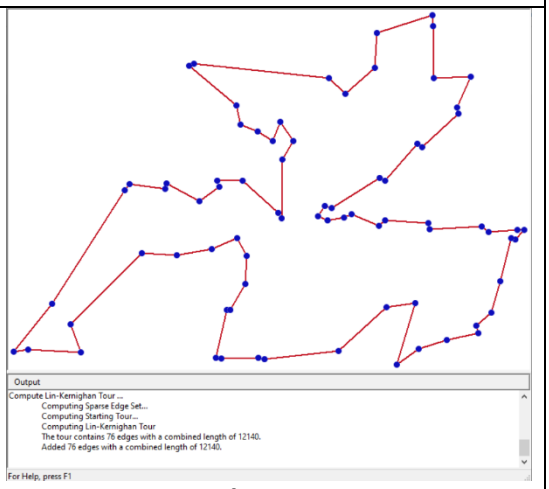
Σειρά: 1.3.3/2.1.1.1/2.1.1.2/2.1.2.1/ 2.1.2.2



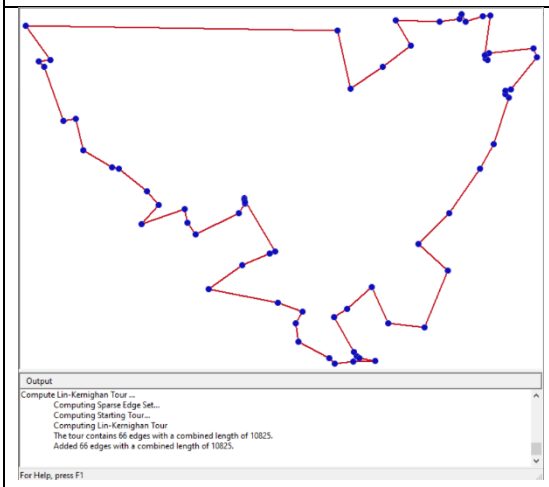
Συστάδα 1.3.3



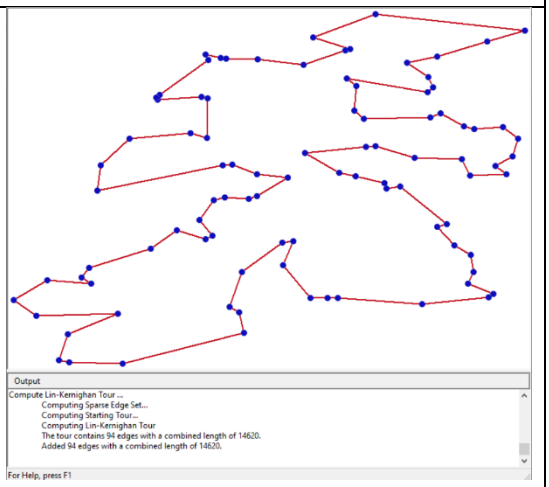
Συστάδα 2.1.1.1



Συστάδα 2.1.1.2



Συστάδα 2.1.2.1

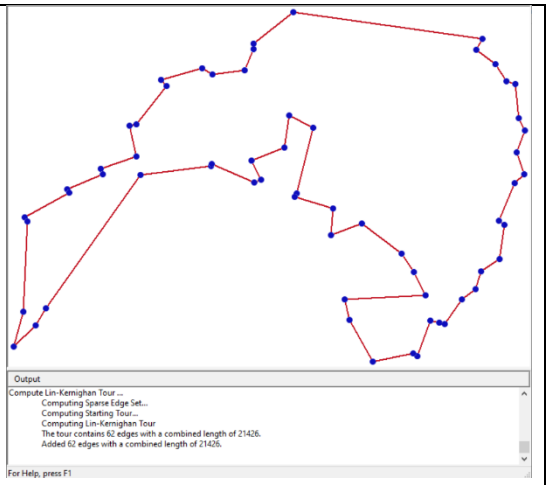


Συστάδα 2.1.2.2

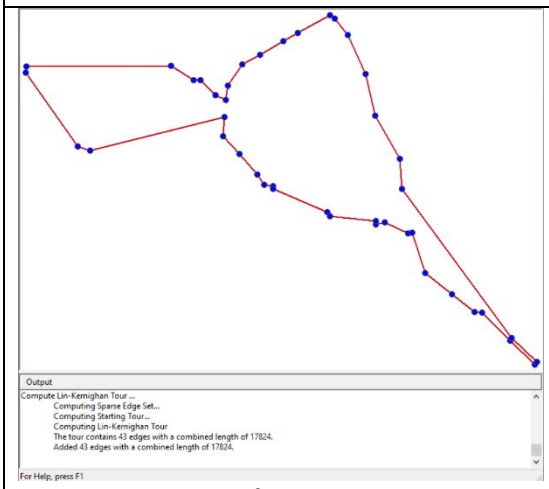
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

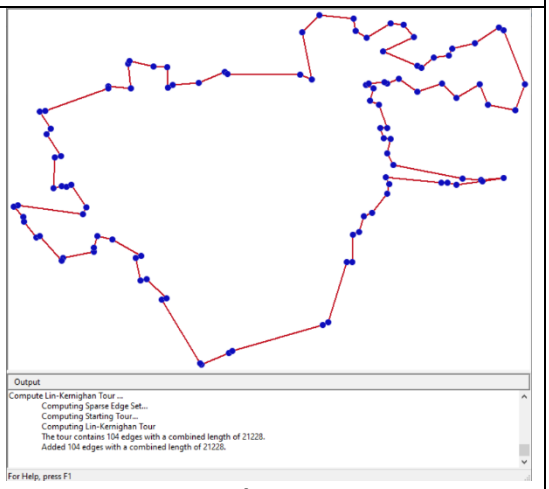
Σειρά: 2.3.1/2.3.2/2.5.1/2.5.2/3.1.1.1



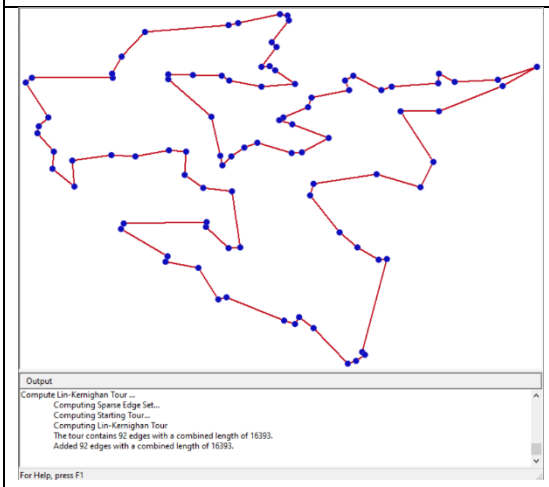
Συστάδα 2.3.1



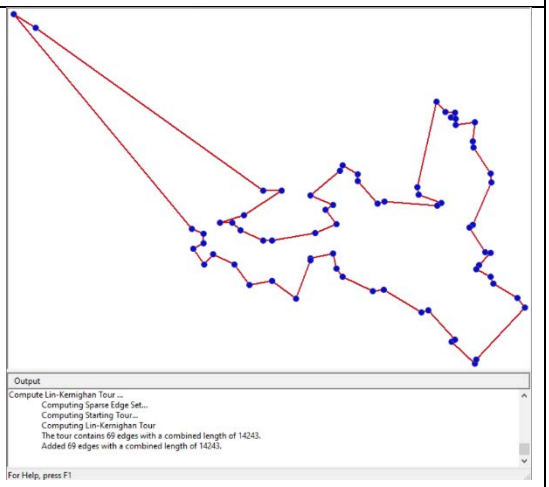
Συστάδα 2.3.2



Συστάδα 2.5.1



Συστάδα 2.5.2

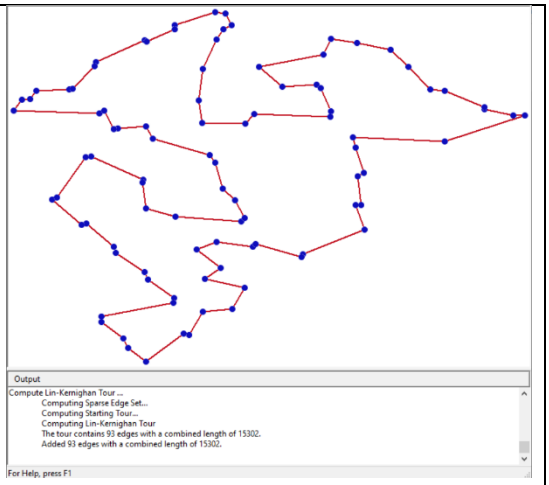


Συστάδα 3.1.1.1

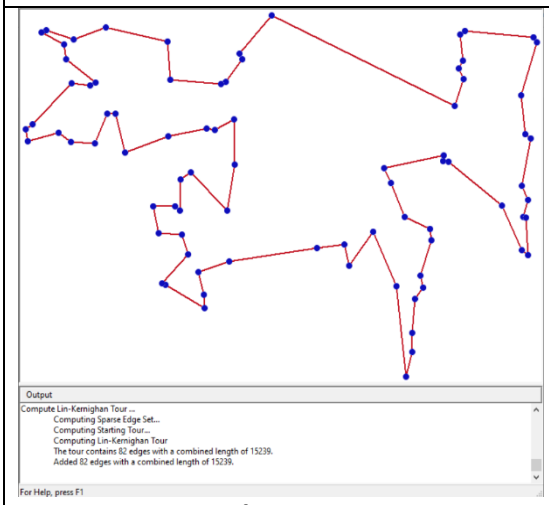
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

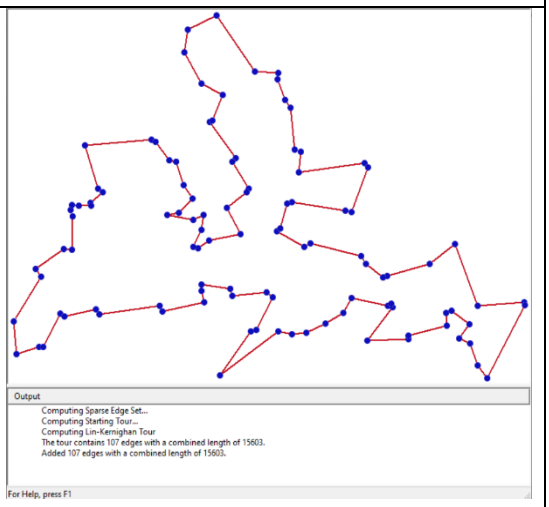
Σειρά: 3.1.1.2/3.1.2.1/3.1.2.2/3.1.3.1/ 3.1.3.2



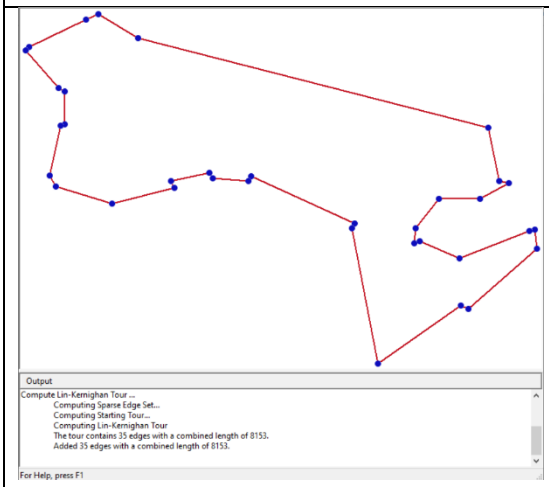
Συστάδα 3.1.1.2



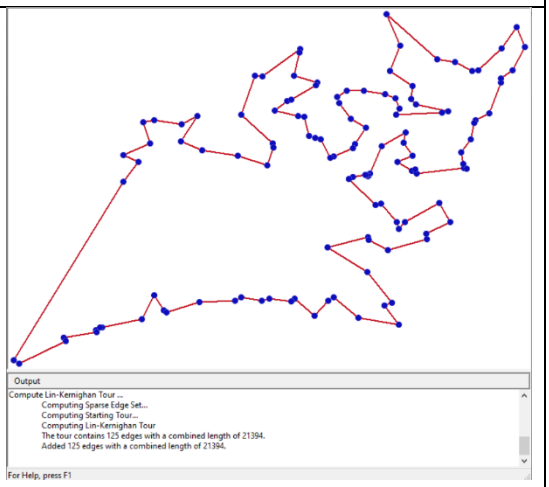
Συστάδα 3.1.2.1



Συστάδα 3.1.2.2



Συστάδα 3.1.3.1

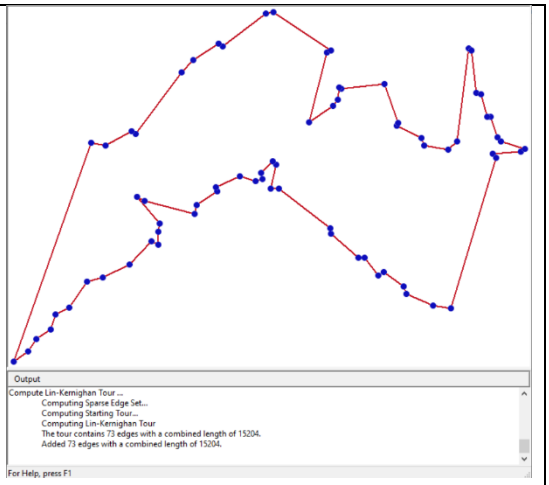


Συστάδα 3.1.3.2

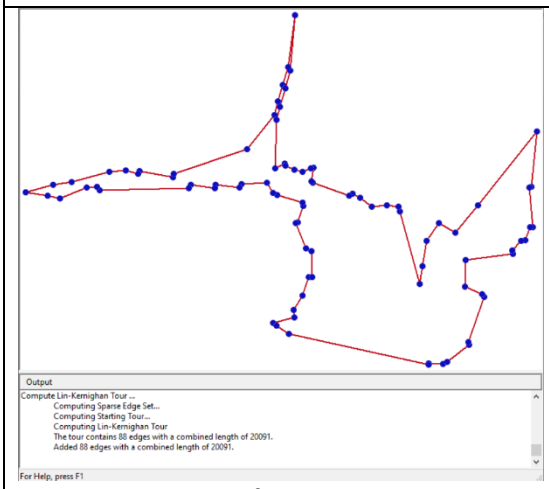
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

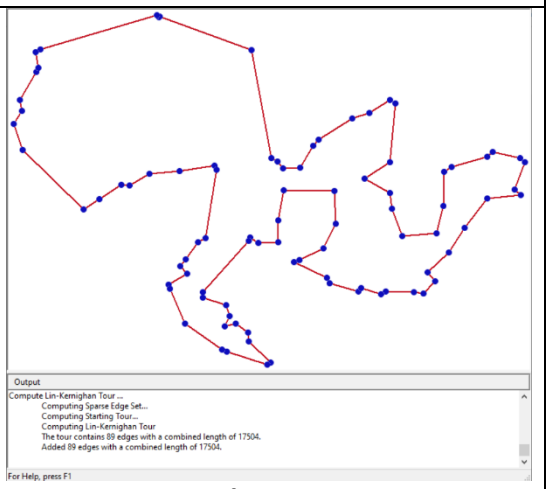
Σειρά: 3.2.1/3.2.2/3.3.1.1/3.3.1.2/ 3.3.2.1



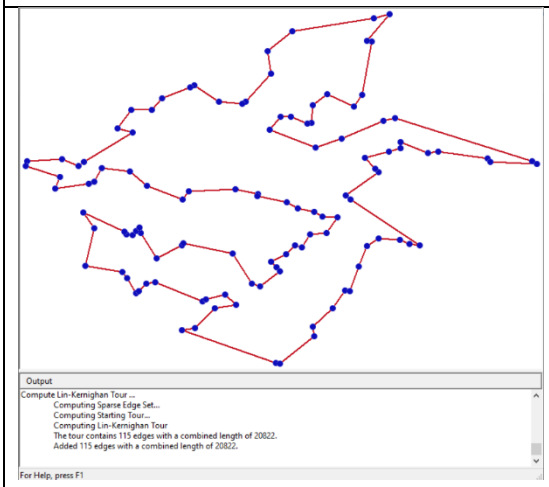
Συστάδα 3.2.1



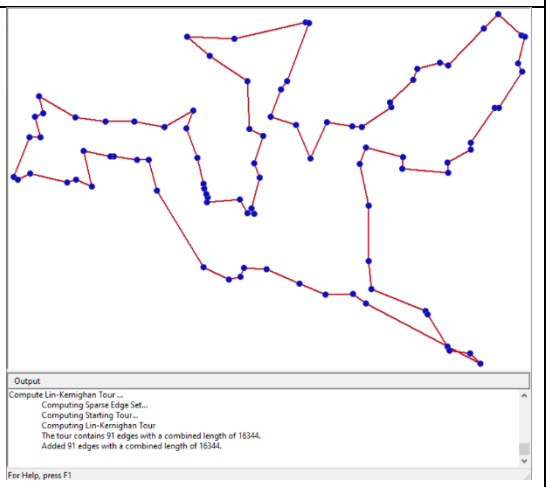
Συστάδα 3.2.2



Συστάδα 3.3.1.1



Συστάδα 3.3.1.2

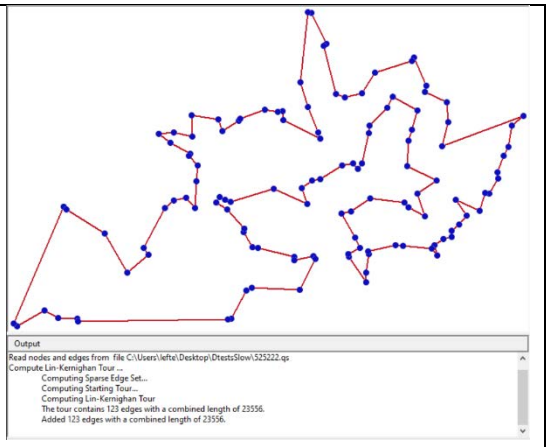


Συστάδα 3.3.2.1

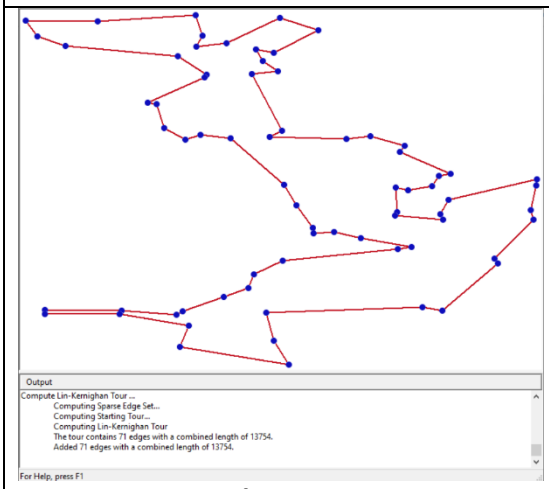
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

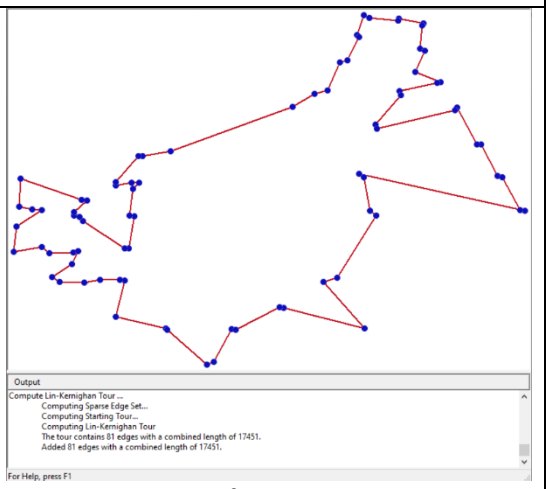
Σειρά: 3.3.2.2/3.4.1.1/3.4.1.2/3.4.2.1/ 3.4.2.2



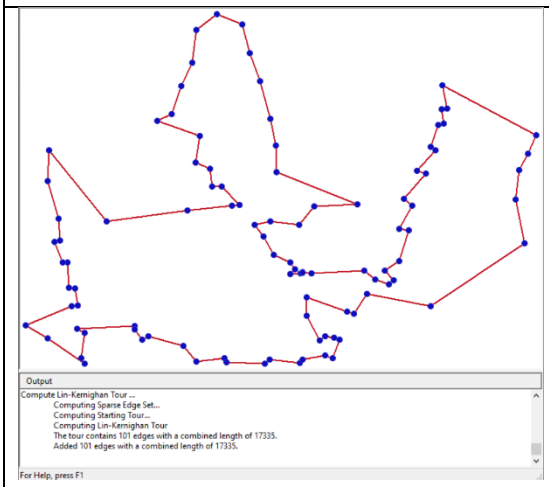
Συστάδα 3.3.2.2



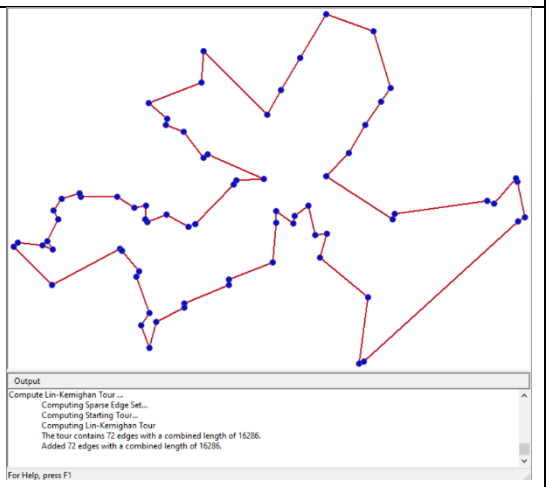
Συστάδα 3.4.1.1



Συστάδα 3.4.1.2



Συστάδα 3.4.2.1

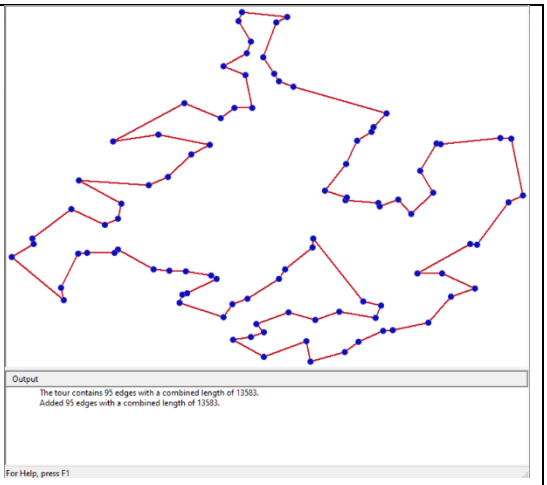


Συστάδα 3.4.2.2

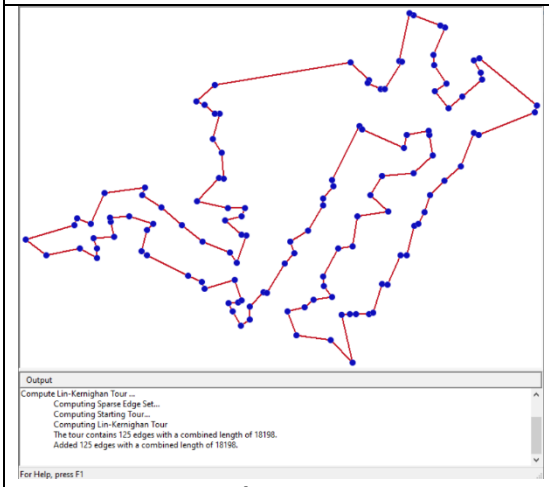
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

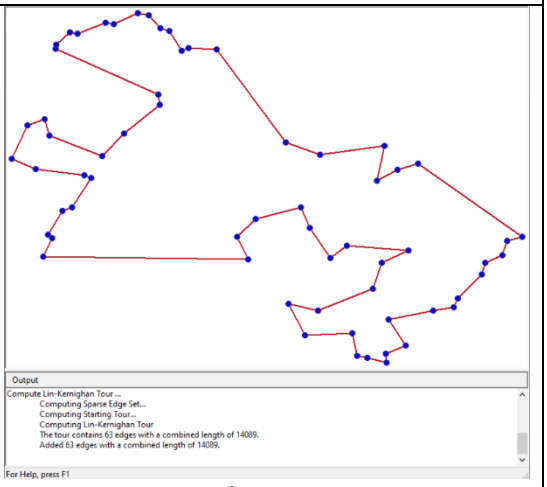
Σειρά: 3.5.1.1/3.5.1.2/3.5.2.1/3.5.2.2/ 3.5.3.1



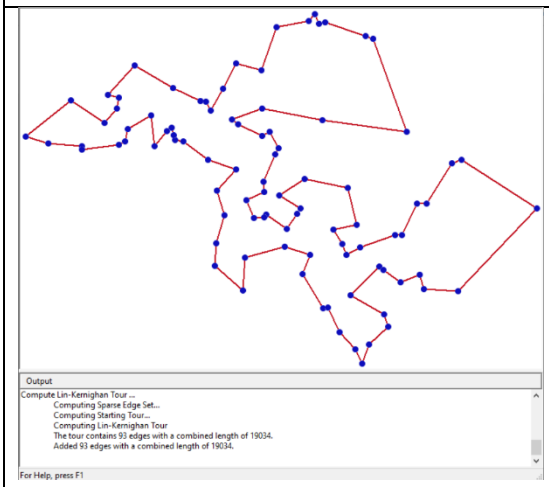
Συστάδα 3.5.1.1



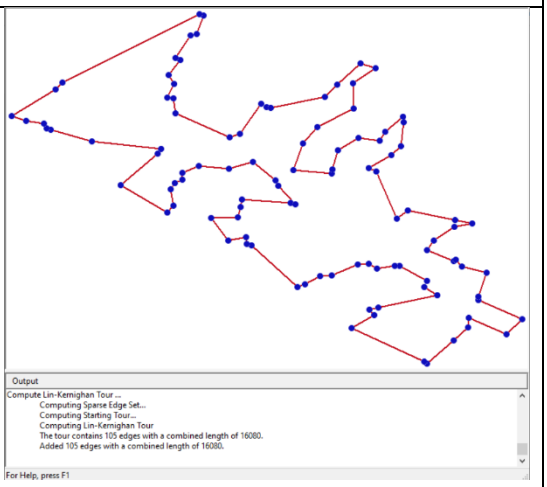
Συστάδα 3.5.1.2



Συστάδα 3.5.2.1



Συστάδα 3.5.2.2

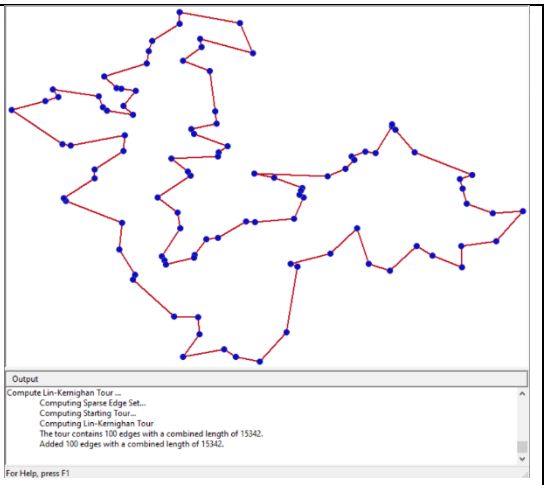


Συστάδα 3.5.3.1

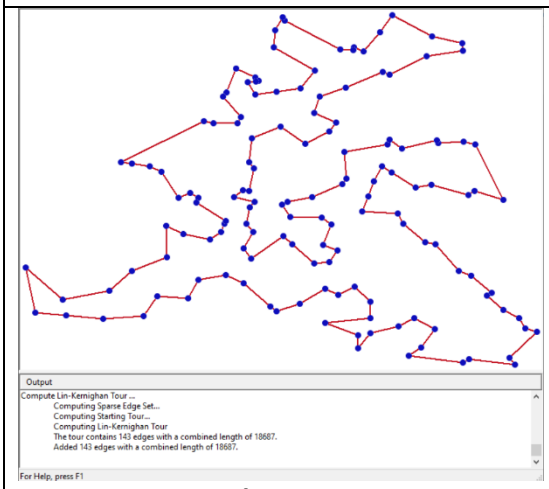
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

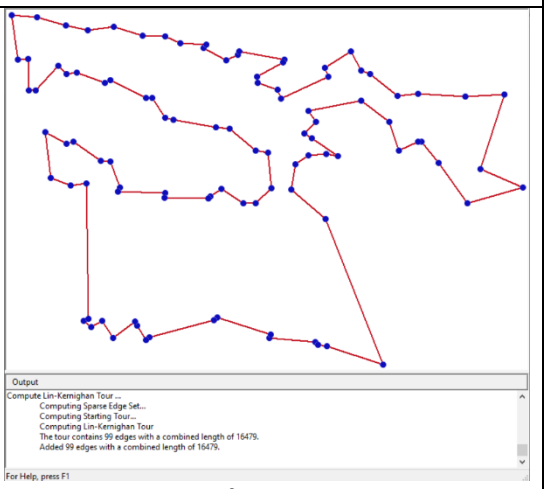
Σειρά: 3.5.3.2/4.1.1.1/4.1.1.2/4.1.2.1/ 4.1.2.2



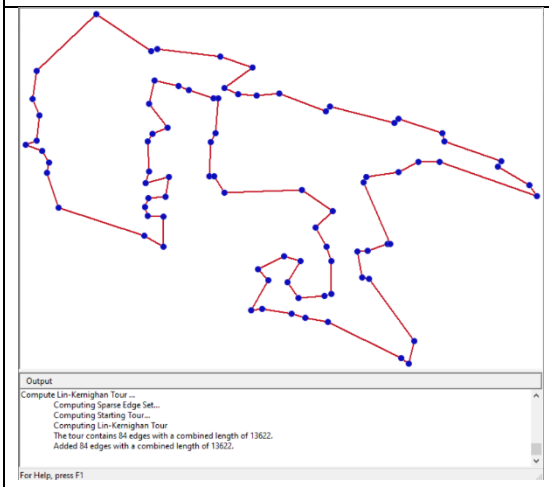
Συστάδα 3.5.3.2



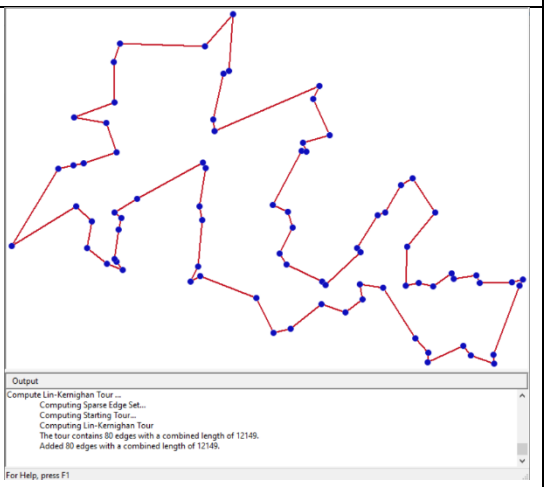
Συστάδα 4.1.1.1



Συστάδα 4.1.1.2



Συστάδα 4.1.2.1

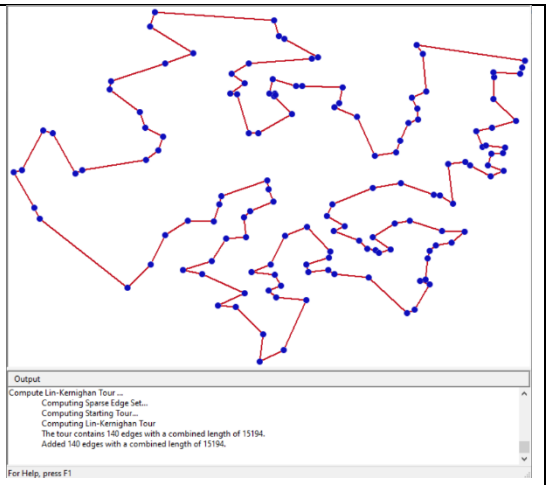


Συστάδα 4.1.2.2

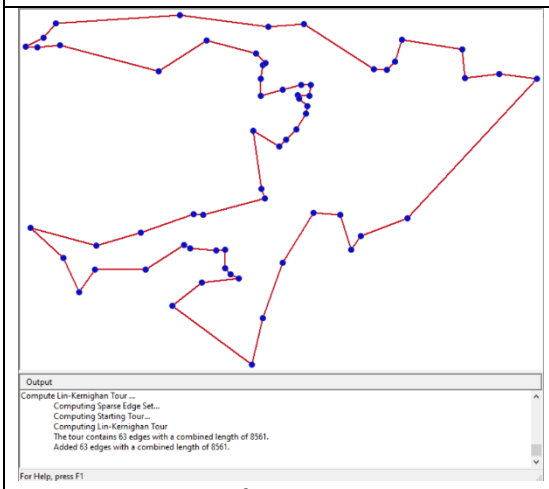
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

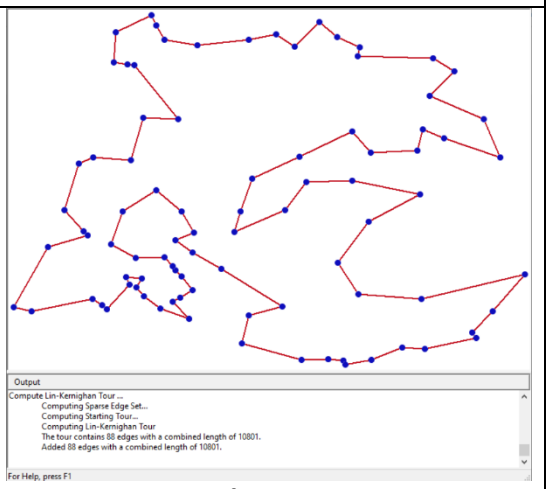
Σειρά: 4.1.3.1/4.1.3.2/4.1.3.3/4.1.4.1/ 4.1.4.2



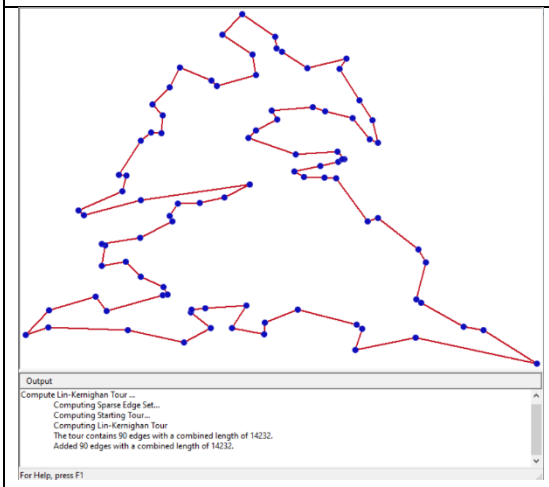
Συστάδα 4.1.3.1



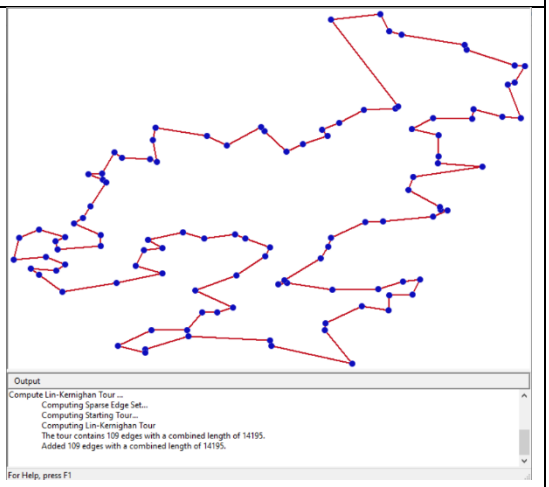
Συστάδα 4.1.3.2



Συστάδα 4.1.3.3



Συστάδα 4.1.4.1

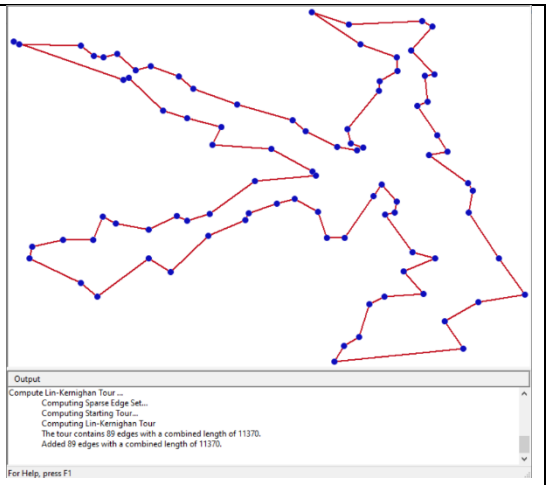


Συστάδα 4.1.4.2

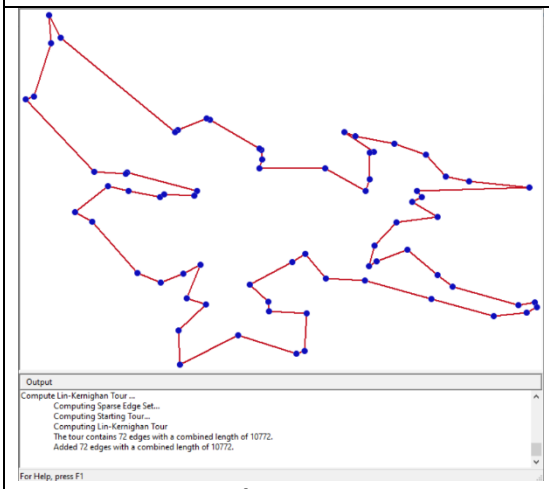
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

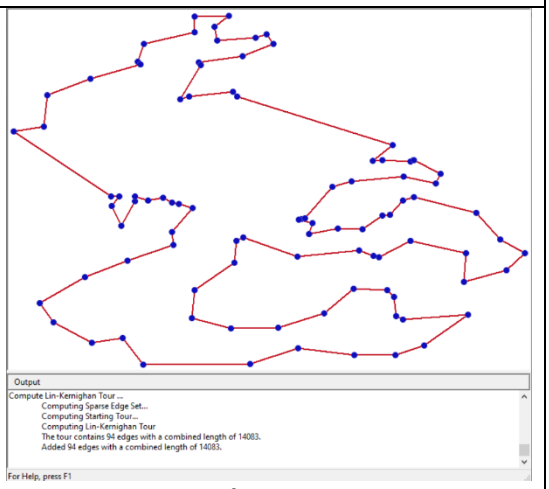
Σειρά: 4.2.1.1/4.2.1.2/4.2.2.1/4.2.2.2/ 4.2.3.1



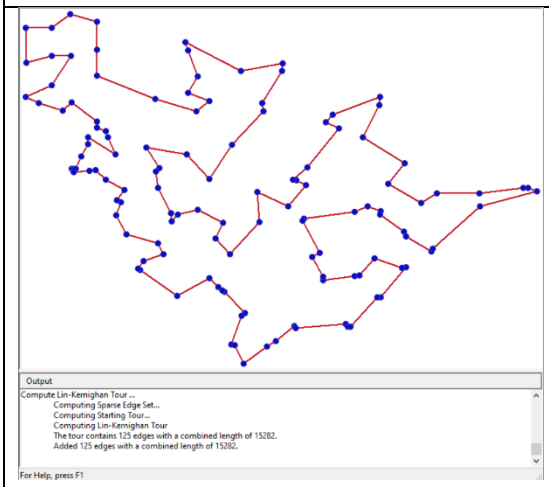
Συστάδα 4.2.1.1



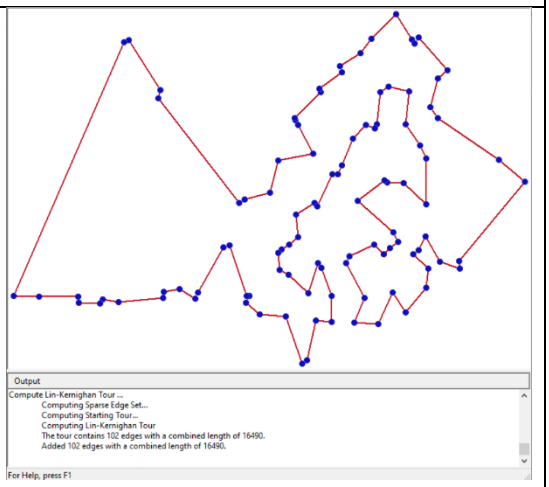
Συστάδα 4.2.1.2



Συστάδα 4.2.2.1



Συστάδα 4.2.2.2

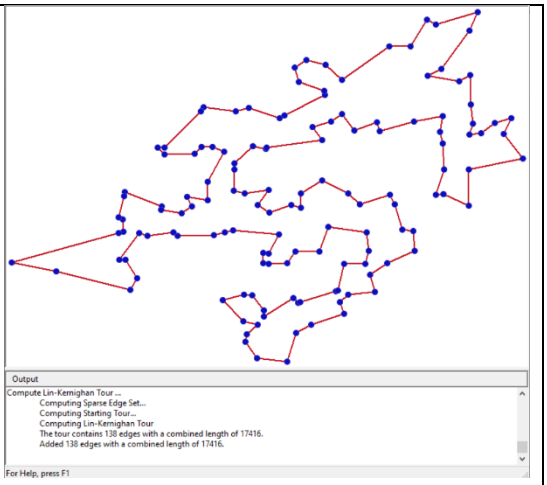


Συστάδα 4.2.3.1

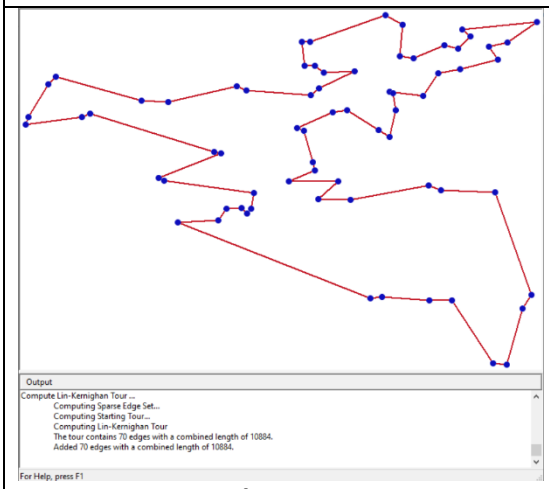
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

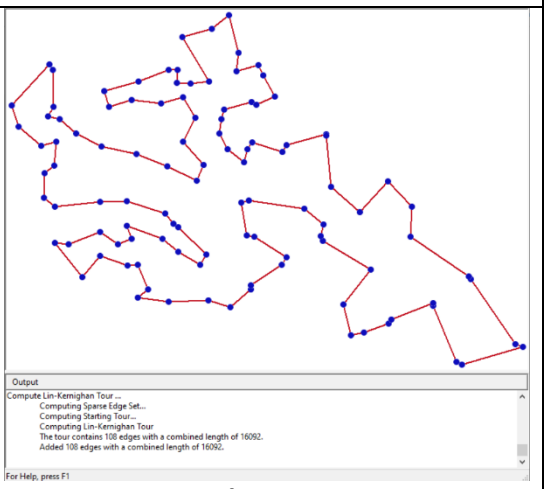
Σειρά: 4.2.3.2/4.2.4.1/4.2.4.2/4.3.1.1/ 4.3.1.2



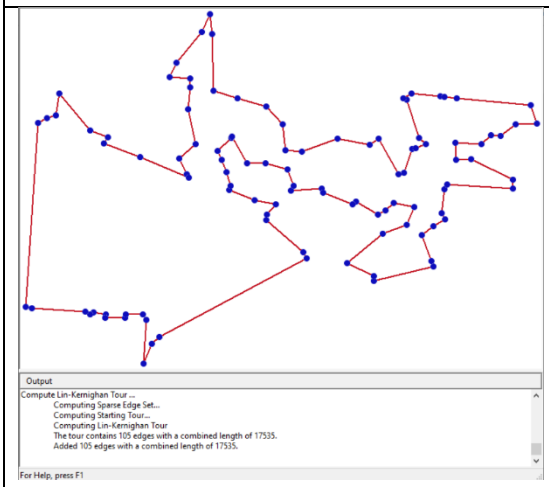
Συστάδα 4.2.3.2



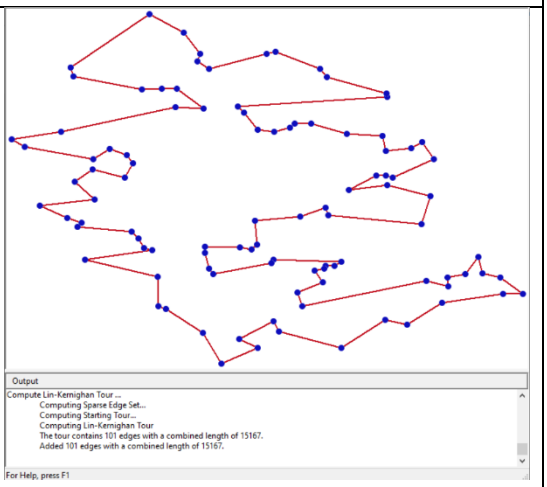
Συστάδα 4.2.4.1



Συστάδα 4.2.4.2



Συστάδα 4.3.1.1

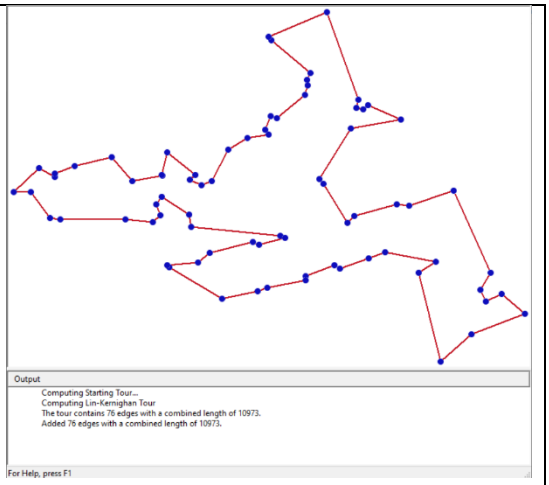


Συστάδα 4.3.1.2

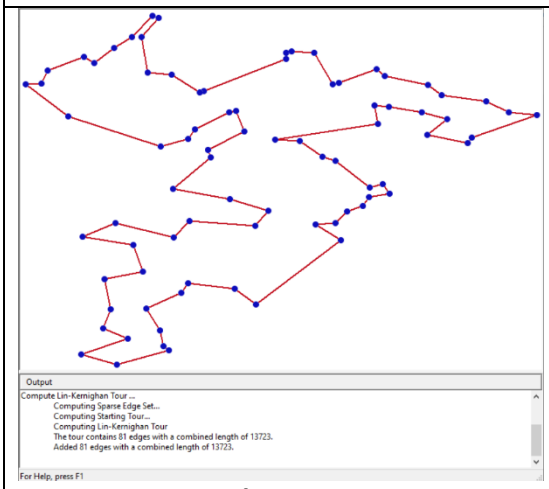
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

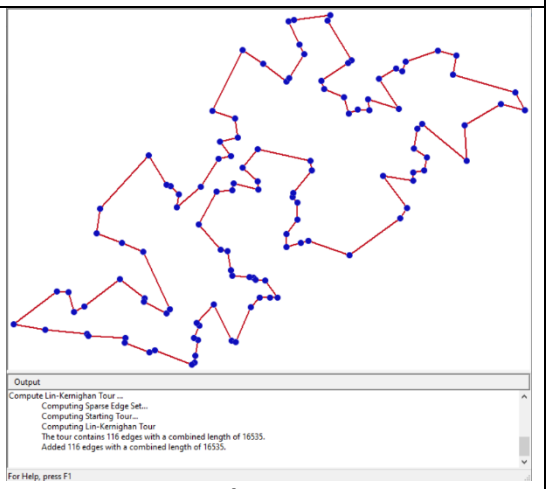
Σειρά: 4.3.1.3/4.3.2.1/4.3.2.2/4.3.3.1/ 4.3.3.2



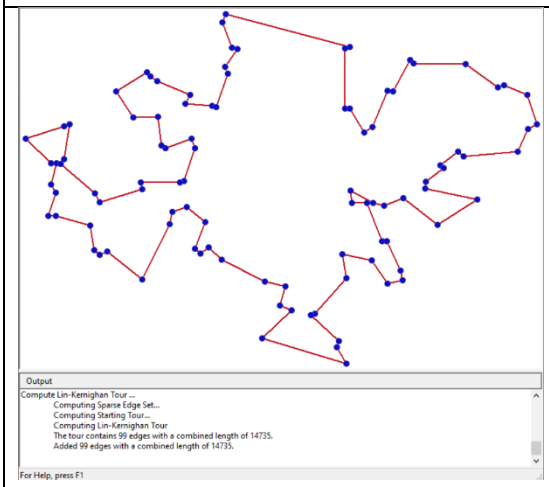
Συστάδα 4.3.1.3



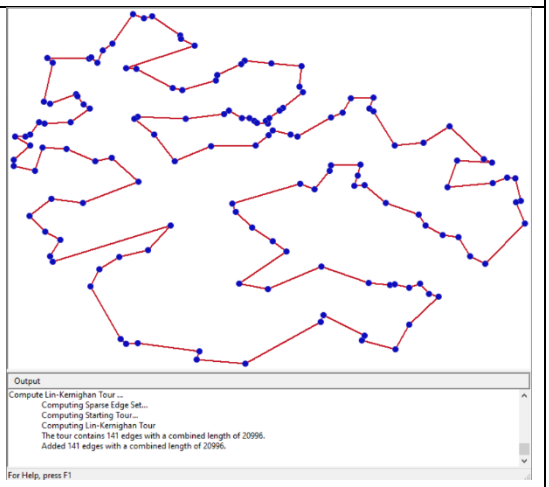
Συστάδα 4.3.2.1



Συστάδα 4.3.2.2



Συστάδα 4.3.3.1

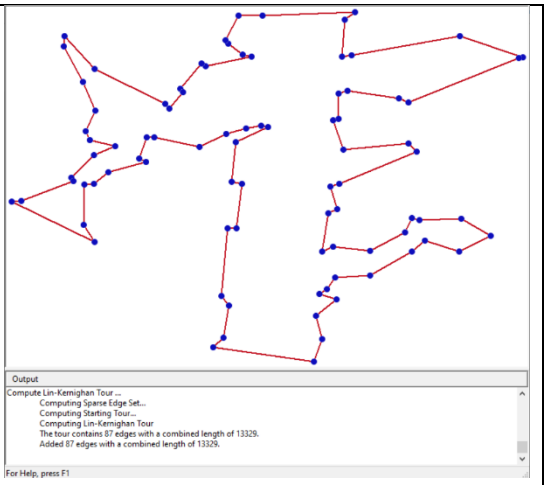


Συστάδα 4.3.3.2

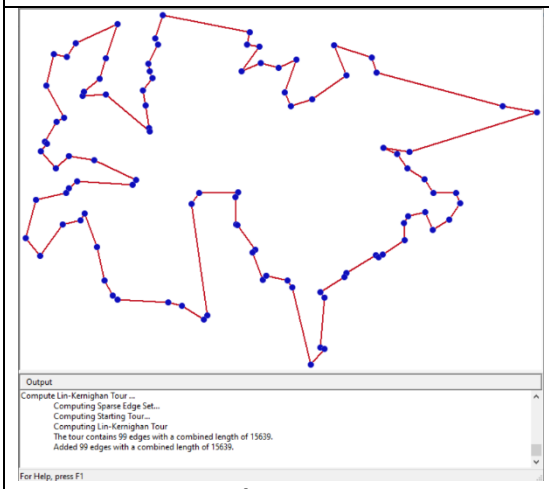
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

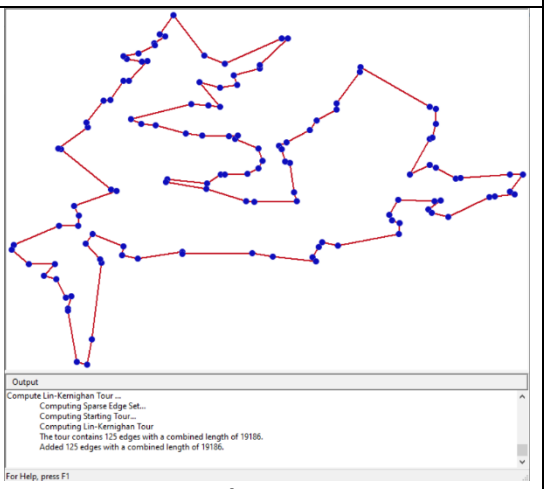
Σειρά: 4.4.1.1/4.4.1.2/4.4.2.1/4.4.2.2/ 4.4.3.1



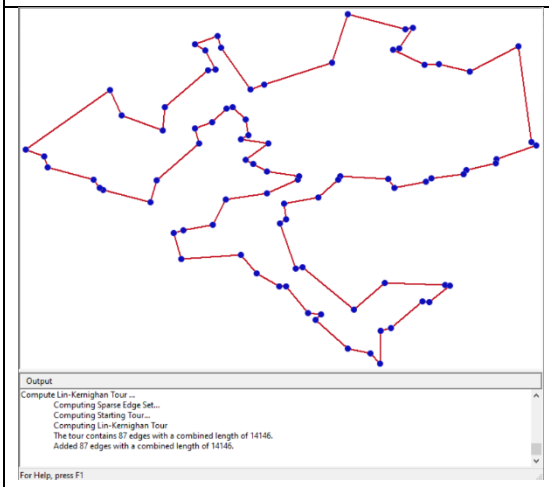
Συστάδα 4.4.1.1



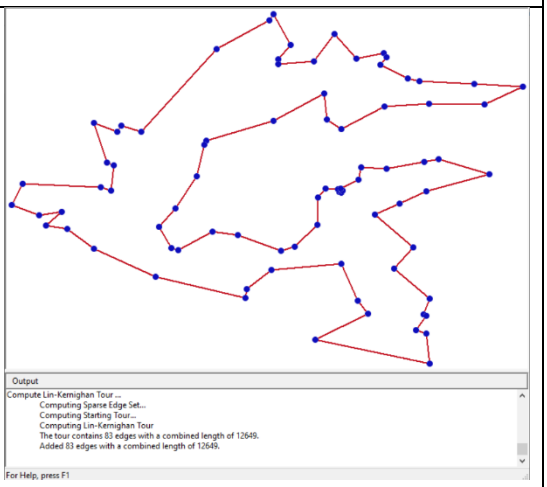
Συστάδα 4.4.1.2



Συστάδα 4.4.2.1



Συστάδα 4.4.2.2

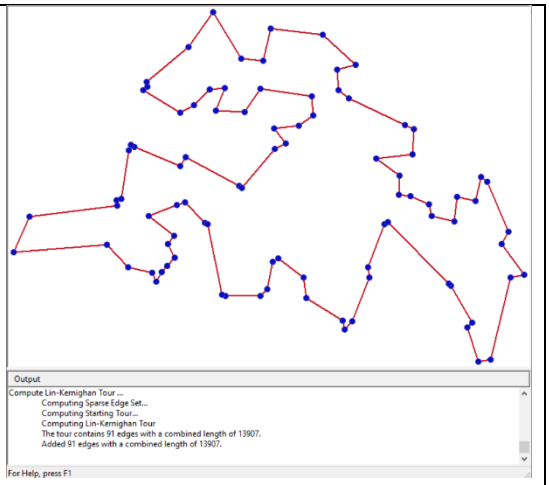


Συστάδα 4.4.3.1

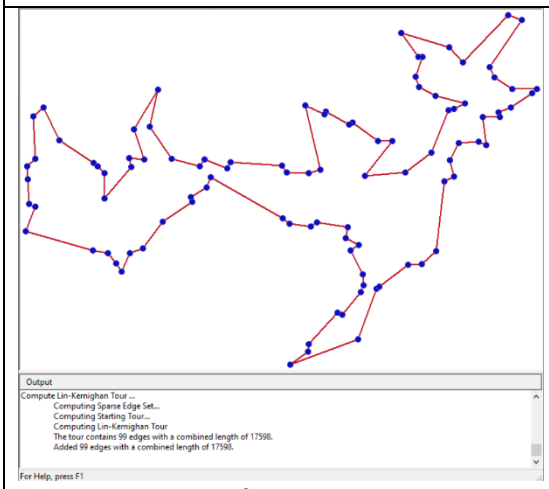
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

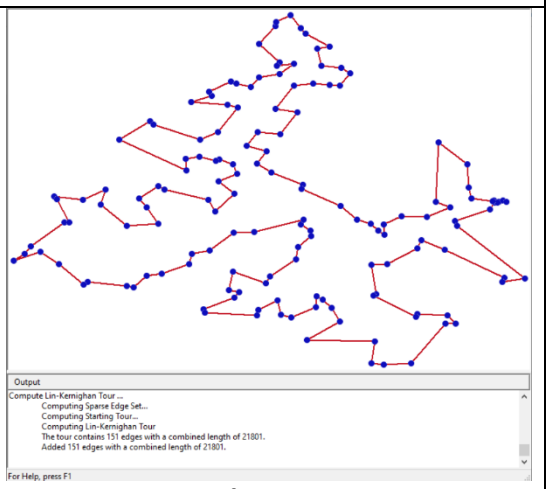
Σειρά: 4.4.3.2/4.5.1.1/4.5.1.2/4.5.2.1/ 4.5.2.2



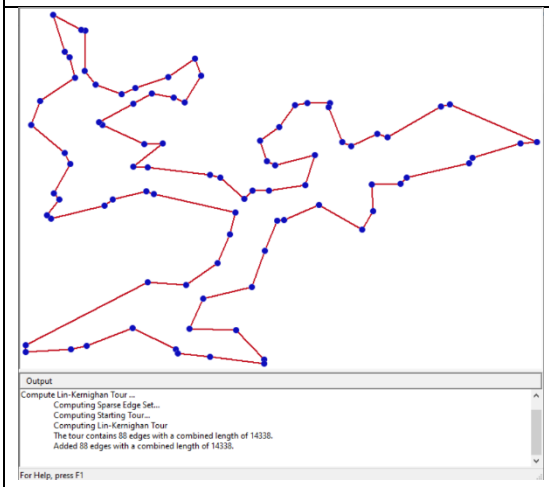
Συστάδα 4.4.3.2



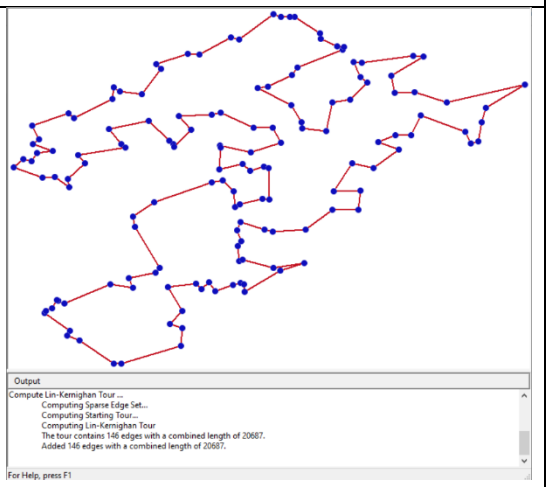
Συστάδα 4.5.1.1



Συστάδα 4.5.1.2



Συστάδα 4.5.2.1

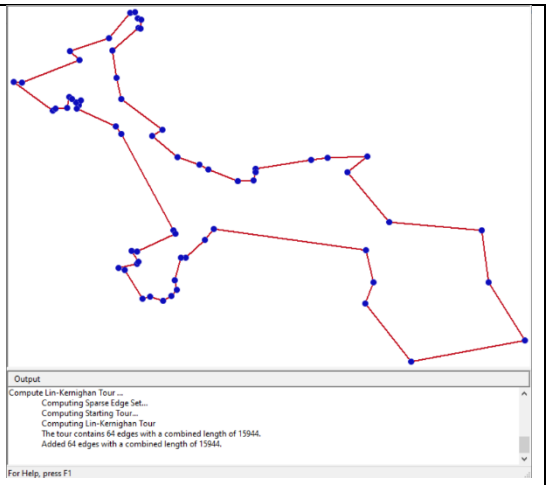


Συστάδα 4.5.2.2

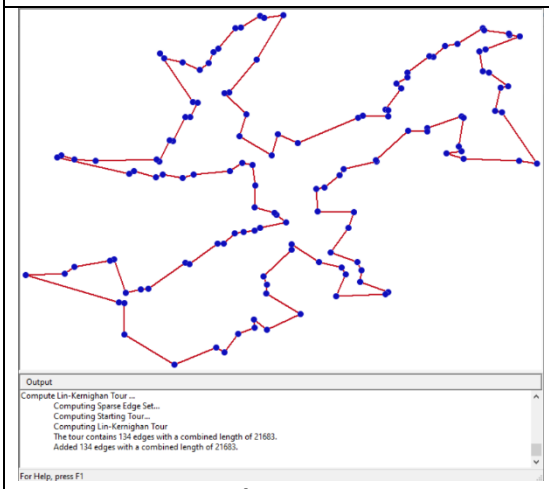
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

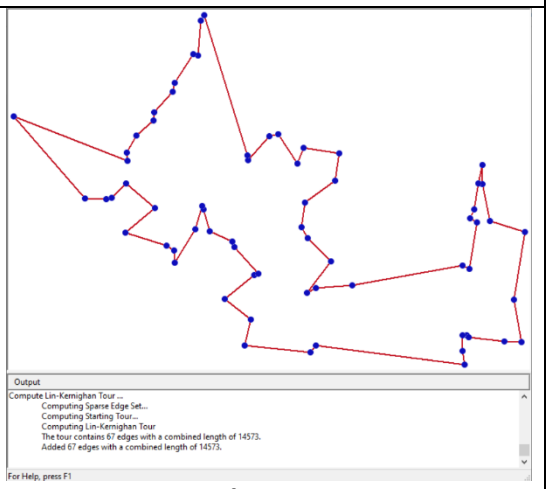
Σειρά: 5.1.1.1/5.1.1.2/5.1.1.3/5.2.1/ 5.2.2



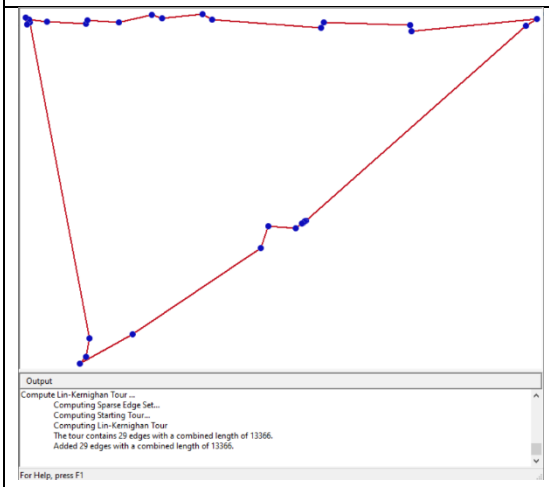
Συστάδα 5.1.1.1



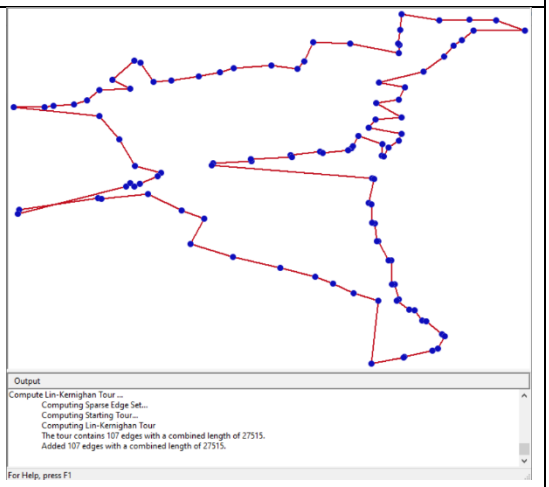
Συστάδα 5.1.1.2



Συστάδα 5.1.1.3



Συστάδα 5.2.1

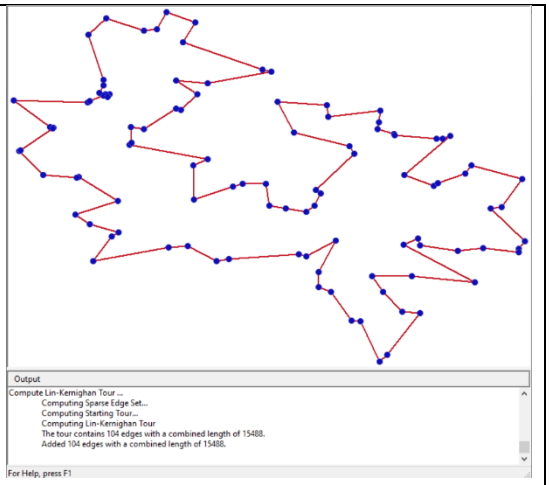


Συστάδα 5.2.2

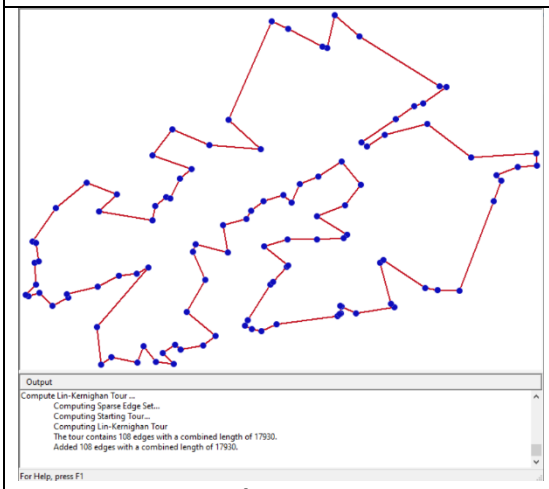
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

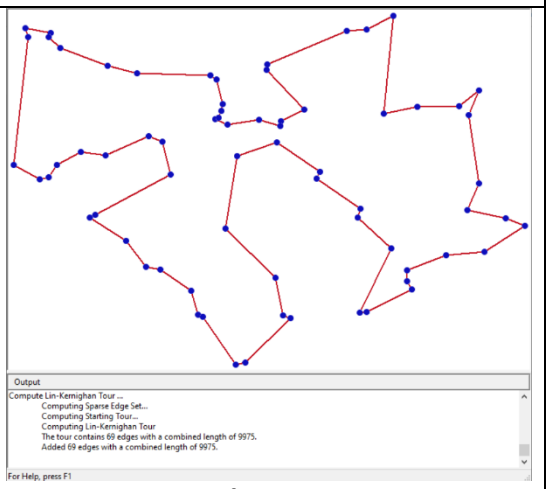
Σειρά: 5.3.1.1/5.3.1.2/5.3.1.3/5.3.2.1/ 5.3.2.2



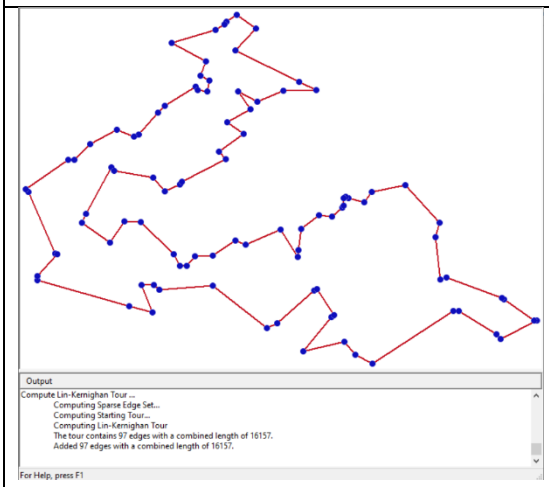
Συστάδα 5.3.1.1



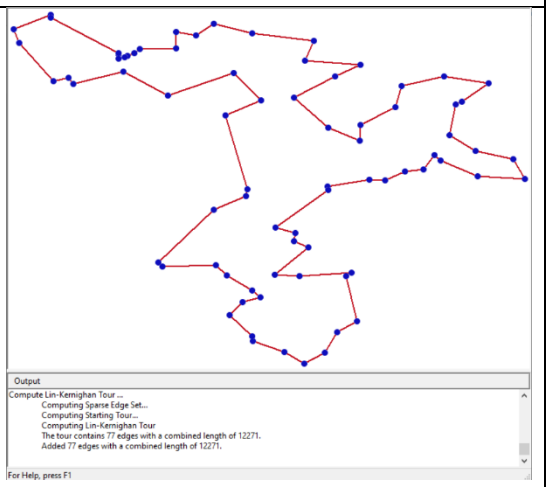
Συστάδα 5.3.1.2



Συστάδα 5.3.1.3



Συστάδα 5.3.2.1

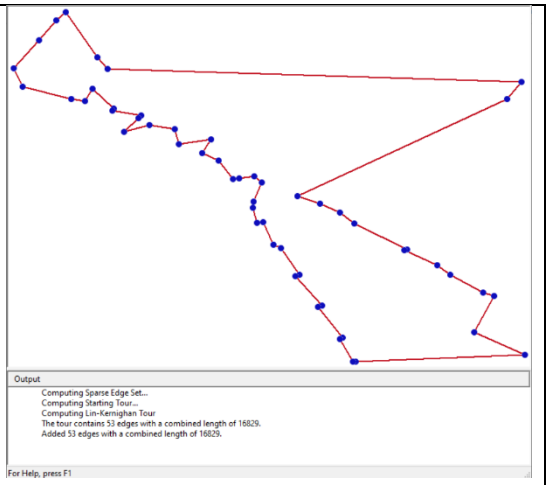


Συστάδα 5.3.2.2

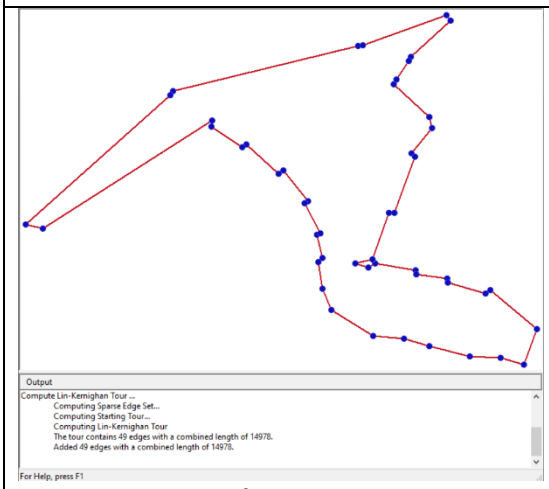
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

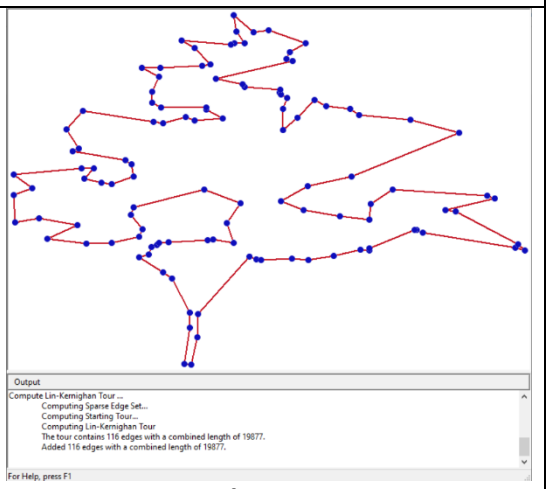
Σειρά: 5.5.1.1/5.5.1.2/5.5.2.1/ 5.5.2.2



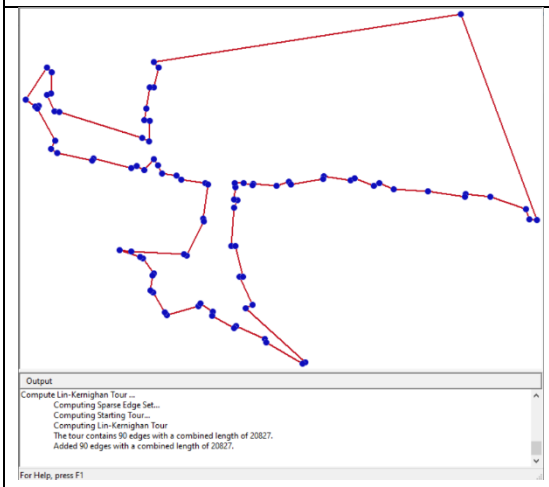
Συστάδα 5.5.1.1



Συστάδα 5.5.1.2



Συστάδα 5.5.2.1

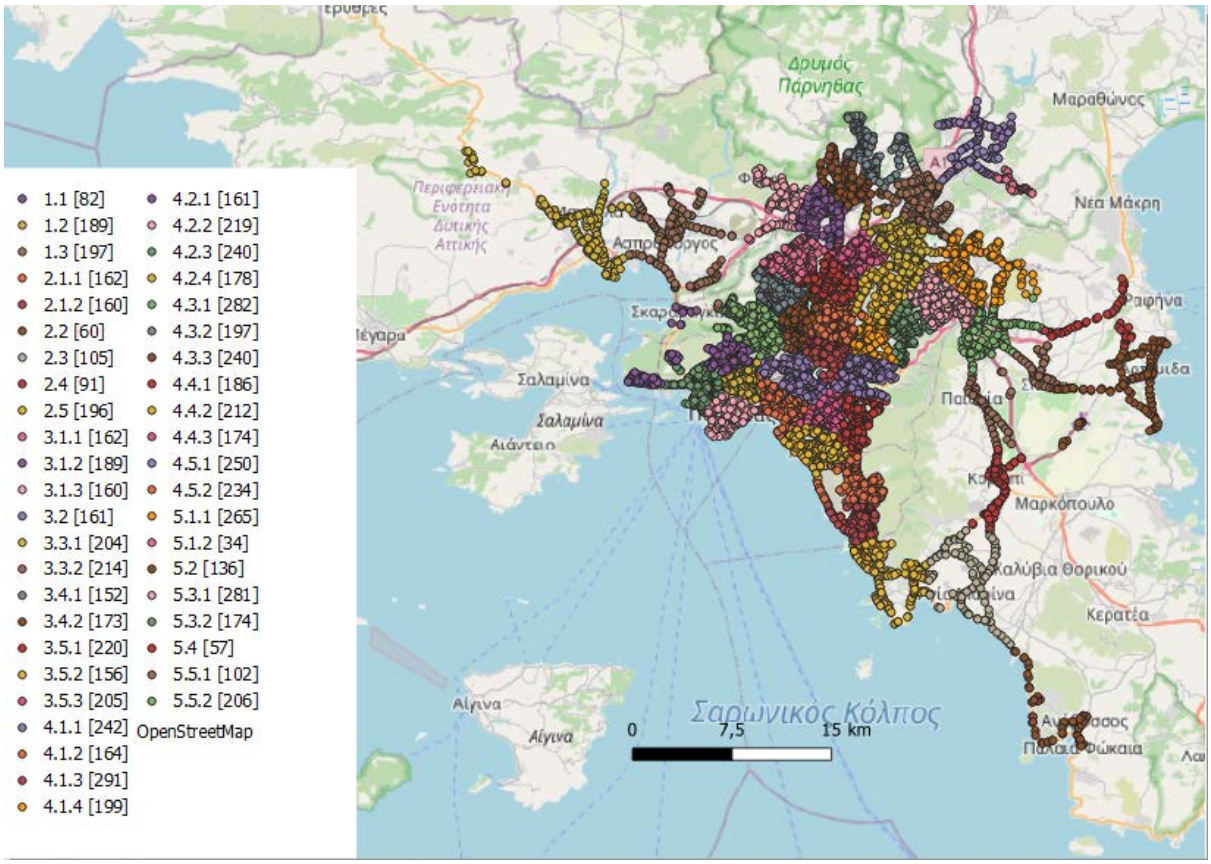


Συστάδα 5.5.2.2

4.6 Παρουσίαση αποτελεσμάτων και παρατηρήσεις

Όπως είδαμε ολοκληρώθηκε η ανάλυση με συνεχή συσταδιοποίηση του δικτύου τόσο για την γρήγορη όσο και για την αργή επίλυση. Καταλήξαμε με αυτή την στρατηγική να χωρίσουμε το δίκτυο μας σε 44 συστάδες για την γρήγορη επίλυση και σε 89 συστάδες για την αργή. Στην συνέχεια παρουσιάζουμε τα αποτελέσματα στα οποία φτάσαμε:

Τελικός χάρτης γρήγορης επίλυσης



Σχήμα 4.14 Συστάδες για γρήγορη επίλυση

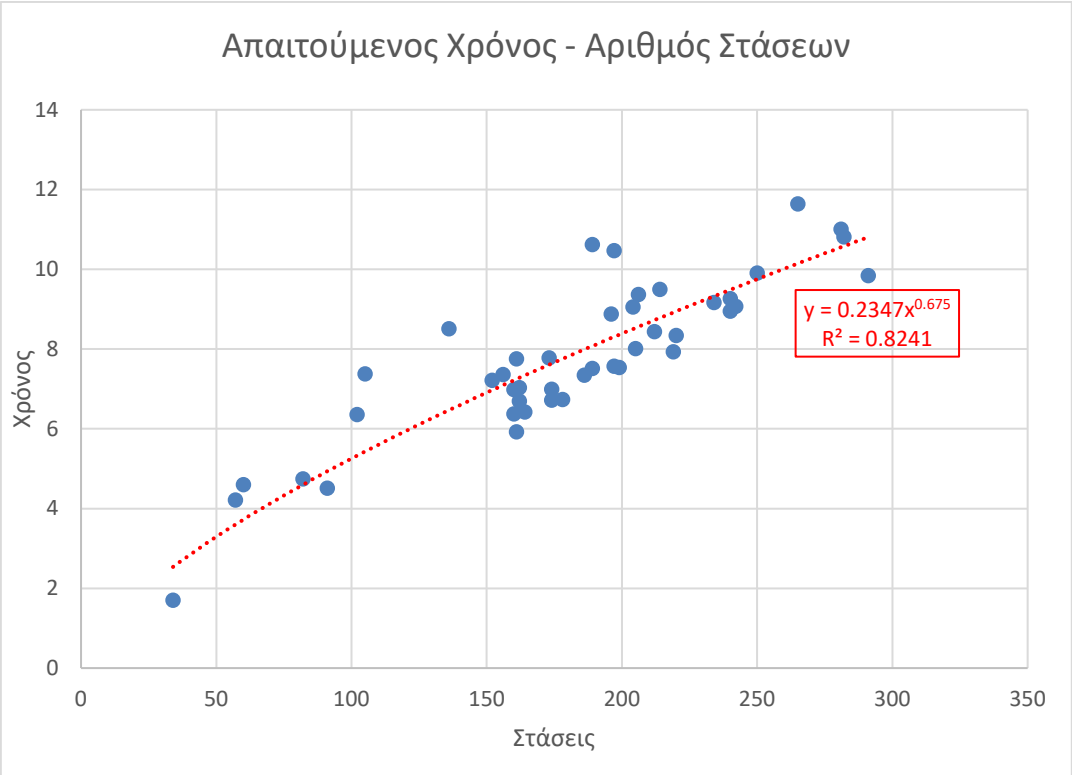
Όπου βλέπουμε την χωρική διάταξη των 44 συστάδων που υποδιαιρέσαμε το δίκτυο μας. Και στον παρακάτω πίνακα βλέπουμε τα αποτελέσματα βέλτιστης δρομολόγησης και συνολικό χρόνο επιθεώρησης:

Πίνακας 4.11 Αποτελέσματα συστάδων γρήγορης επίλυσης

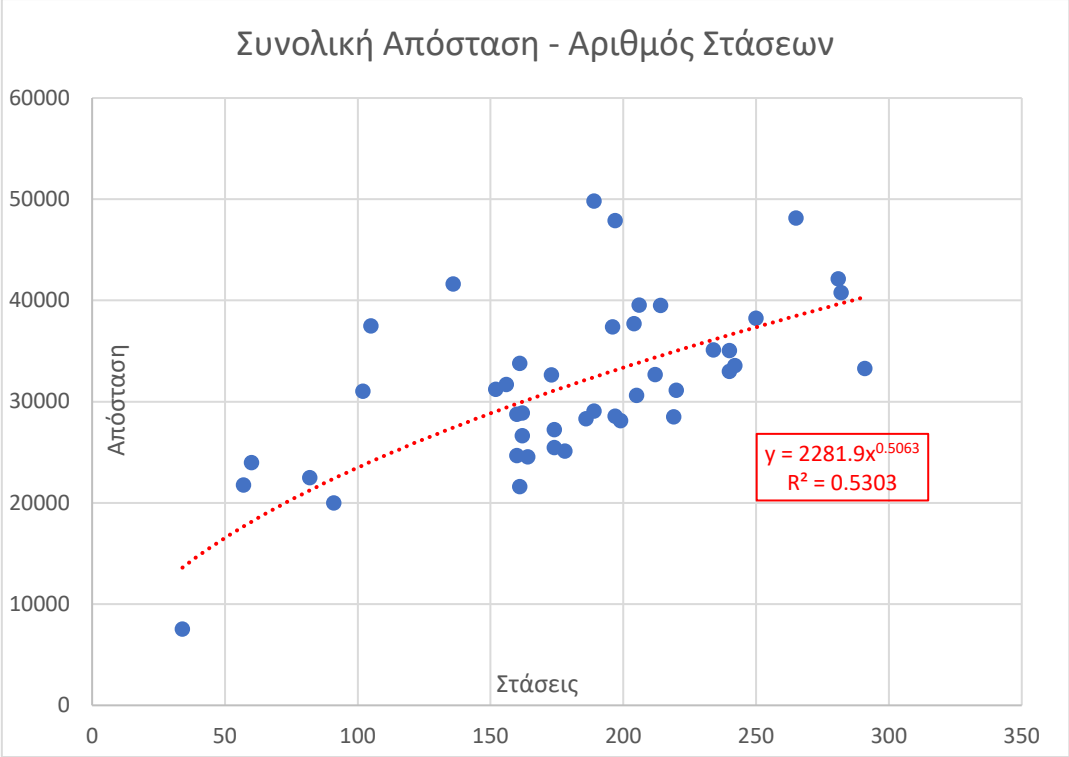
Συστάδα	*Στάσεις*	*Απόσταση*	*Χρόνος*
1.1	82	22504	4.742267
1.2	189	49803	10.62045
1.3	197	47884	10.46593
2.1.1	162	26650	6.6975
2.1.2	160	24691	6.370317
2.2	60	23995	4.59925
2.3	105	37487	7.37305
2.4	91	19981	4.513817
2.5	196	37403	8.877117
3.1.1	162	28884	7.0326
3.1.2	189	29069	7.51035
3.1.3	160	28749	6.979017
3.2	161	33779	7.750183
3.3.1	204	37705	9.05575
3.3.2	214	39514	9.493767
3.4.1	152	31207	7.214383
3.4.2	173	32631	7.777983
3.5.1	220	31130	8.336167
3.5.2	156	31711	7.35665
3.5.3	205	30620	8.009667
4.1.1	242	33574	9.069433
4.1.2	164	24564	6.417933
4.1.3	291	33281	9.84215
4.1.4	199	28115	7.533917
4.2.1	161	21596	5.922733
4.2.2	219	28517	7.92755
4.2.3	240	33005	8.95075
4.2.4	178	25113	6.733617
4.3.1	282	40771	10.81565
4.3.2	197	28572	7.569133
4.3.3	240	35062	9.2593
4.4.1	186	28315	7.34725
4.4.2	212	32672	8.434133
4.4.3	174	25474	6.7211
4.5.1	250	38228	9.900867
4.5.2	234	35098	9.1647
5.1.1	265	48130	11.63617
5.1.2	34	7534	1.696767
5.2	136	41616	8.509067
5.3.1	281	42129	11.00268
5.3.2	174	27245	6.98675

5.4	57	21754	4.2131
5.5.1	102	31037	6.35555
5.5.2	206	39532	9.363133

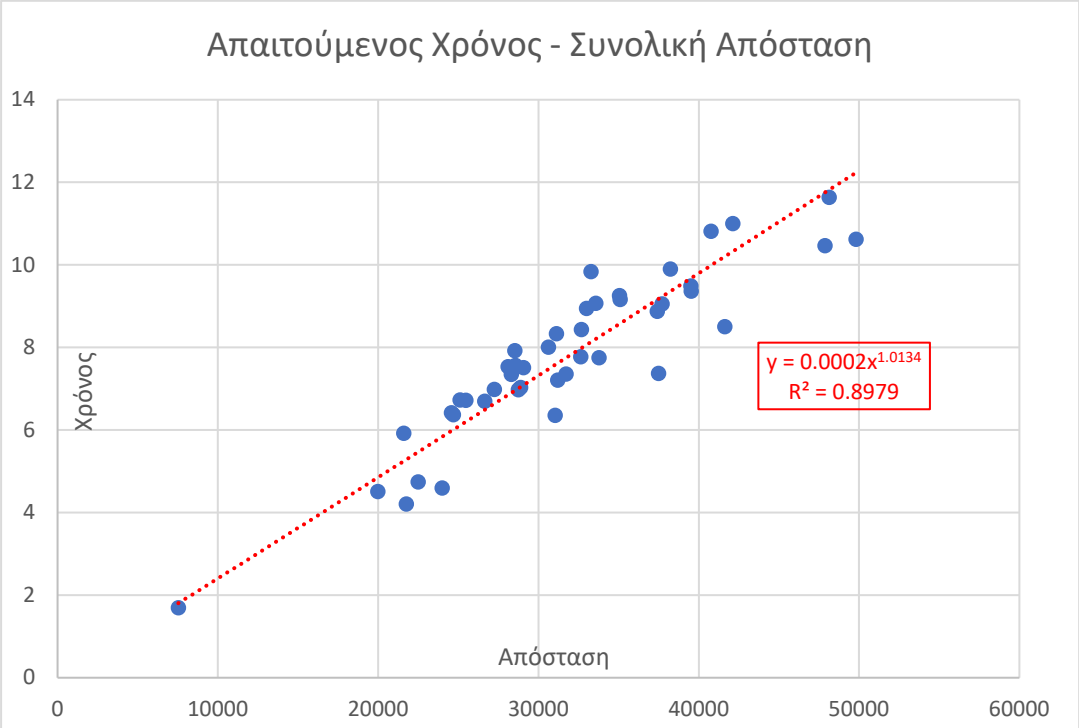
Από την ανάλυση παίρνουμε τα εξής σημαντικά διαγράμματα:



Εδώ βλέπουμε πως ο αριθμός των στάσεων επηρεάζει τον συνολικό χρόνο επιθεώρησης που απαιτείται. Μοντελοποιήσαμε τα σημεία του γραφήματος μας με την εξίσωση $y = 0.2347x^{0.675}$ καθώς ήταν η εξίσωση στην γκάμα των εξισώσεων που προσφέρει το Excel για μοντελοποίηση που εμφάνιζε το μικρότερο R – squared Value δηλαδή ήταν η εξίσωση που ταίριαζε καλυτέρα να περιγράψει τα πειραματικά μας δεδομένα.



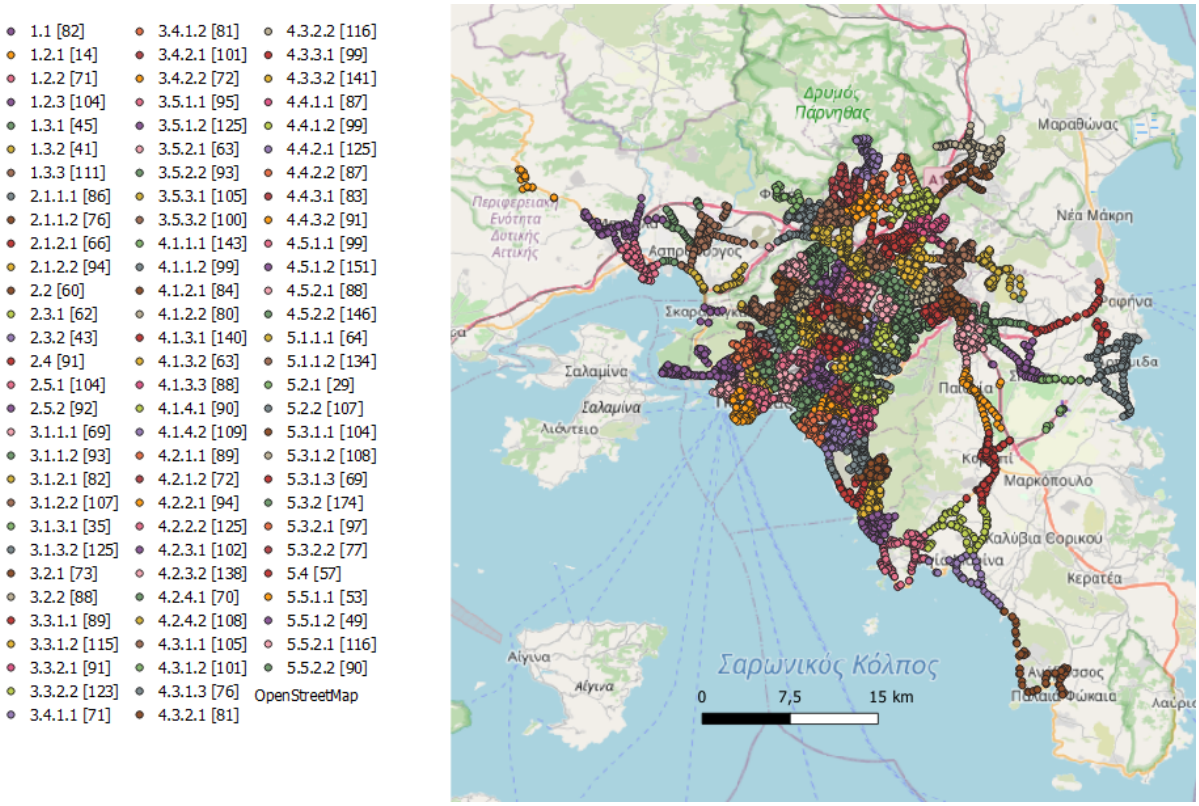
Στην συνέχεια έχουμε το διάγραμμα που μας δίνει μια αντίληψη στο πως το σύνολο των στάσεων που έχει η κάθε συστάδα αυξάνει την συνολική απόσταση που πρέπει να καλυφθεί για να ολοκληρωθεί η επιθεώρηση. Ακολουθήσαμε όμοια λογική με πριν για την μοντελοποίηση των πειραματικών δεδομένων και καταλήξαμε στην εξίσωση $y = 2281.9x^{0.5063}$.



Τέλος έχουμε το διάγραμμα που εμφανίζει πως η απόσταση της βέλτιστης διαδρομής επηρεάζει τον χρόνο που απαιτείται για την επιθεώρηση. Παρατηρούμε με το μάτι ότι από την σύγκριση των δεδομένων μας αυτό μοιάζει να είναι που ακολουθεί την πιο γραμμική συμπεριφορά, ωστόσο η καλύτερη εξίσωση που βρήκαμε να περιγράφει τα δεδομένα μας είναι $y = 0.0002x^{1.0134}$.

Προχωράμε στην ανάλυση των αποτελεσμάτων της αργής επίλυσης.

Τελικός χάρτης αργής επίλυσης



Σχήμα 4.15 Συστάδες για αργή επίλυση

Όπου βλέπουμε την χωρική διάταξη των 89 συστάδων που υποδιαιρέσαμε το δίκτυο μας. Και στον παρακάτω πίνακα βλέπουμε τα αποτελέσματα βέλτιστης δρομολόγησης και συνολικό χρόνο επιθεώρησης:

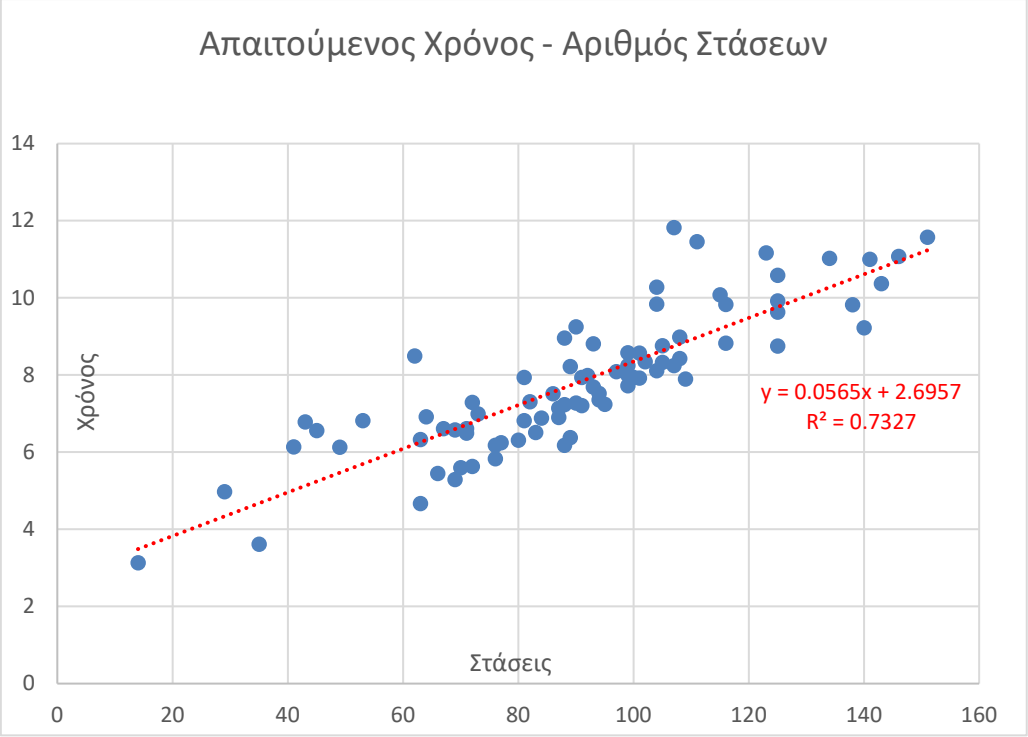
Πίνακας 4.12 Αποτελέσματα συστάδων αργής επίλυσης

Συστάδα	Στάσεις	Απόσταση	Χρόνος
1.1	82	22504	9.484533
1.2.1	14	8872	3.128267
1.2.2	71	14130	6.605667
1.2.3	104	22705	10.27817

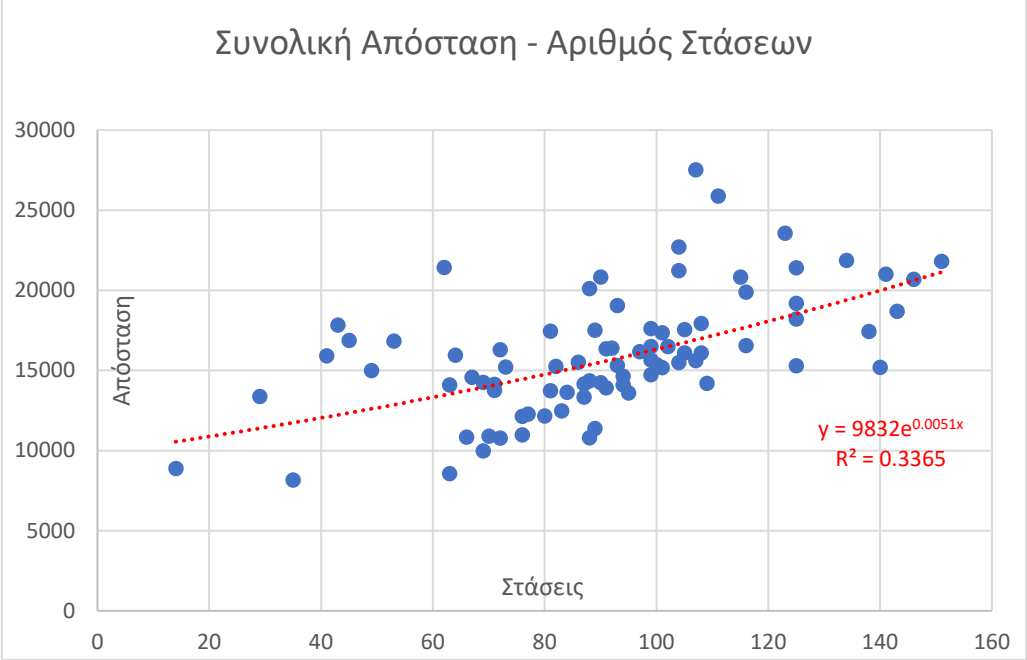
1.3.1	45	16872	6.5616
1.3.2	41	15910	6.139667
1.3.3	111	25864	11.4592
2.1.1.1	86	15501	7.516967
2.1.1.2	76	12140	6.175333
2.1.2.1	66	10825	5.4475
2.1.2.2	94	14620	7.519333
2.2	60	23995	9.1985
2.3.1	62	21426	8.494467
2.3.2	43	17824	6.780533
2.4	91	19981	9.027633
2.5.1	104	21228	9.835067
2.5.2	92	16393	7.984567
3.1.1.1	69	14243	6.5729
3.1.1.2	93	15302	7.6906
3.1.2.1	82	15239	7.305033
3.1.2.2	107	15603	8.247567
3.1.3.1	35	8153	3.612567
3.1.3.2	125	21394	10.58487
3.2.1	73	15204	6.994533
3.2.2	88	20091	8.960633
3.3.1.1	89	17504	8.217867
3.3.1.2	115	20822	10.07993
3.3.2.1	91	16344	7.936533
3.3.2.2	123	23556	11.1668
3.4.1.1	71	13754	6.492867
3.4.1.2	81	17451	7.9353
3.4.2.1	101	17335	8.567167
3.4.2.2	72	16286	7.2858
3.5.1.1	95	13583	7.241567
3.5.1.2	125	18198	9.626067
3.5.2.1	63	14089	6.3267
3.5.2.2	93	19034	8.8102
3.5.3.1	105	16080	8.324
3.5.3.2	100	15342	7.935933
4.1.1.1	143	18687	10.37277
4.1.1.2	99	16479	8.2437
4.1.2.1	84	13622	6.8866
4.1.2.2	80	12149	6.311367
4.1.3.1	140	15194	9.224867
4.1.3.2	63	8561	4.6683
4.1.3.3	88	10801	6.173633
4.1.4.1	90	14232	7.2696
4.1.4.2	109	14195	7.891833
4.2.1.1	89	11370	6.377667
4.2.1.2	72	10772	5.6316

4.2.2.1	94	14083	7.358233
4.2.2.2	125	15282	8.751267
4.2.3.1	102	16490	8.347
4.2.3.2	138	17416	9.8248
4.2.4.1	70	10884	5.598533
4.2.4.2	108	16092	8.4276
4.3.1.1	105	17535	8.7605
4.3.1.2	101	15167	7.916767
4.3.1.3	76	10973	5.825233
4.3.2.1	81	13723	6.8169
4.3.2.2	116	16535	8.827167
4.3.3.1	99	14735	7.7205
4.3.3.2	141	20996	10.9988
4.4.1.1	87	13329	6.8987
4.4.1.2	99	15639	7.9917
4.4.2.1	125	19186	9.922467
4.4.2.2	87	14146	7.1438
4.4.3.1	83	12469	6.507367
4.4.3.2	91	13907	7.205433
4.5.1.1	99	17598	8.5794
4.5.1.2	151	21801	11.57363
4.5.2.1	88	14338	7.234733
4.5.2.2	146	20687	11.07277
5.1.1.1	64	15944	6.916533
5.1.1.2	134	21863	11.02557
5.1.1.3	67	14573	6.605233
5.1.2	34	7534	3.393533
5.2.1	29	13366	4.976467
5.2.2	107	27515	11.82117
5.3.1.1	104	15488	8.113067
5.3.1.2	108	17930	8.979
5.3.1.3	69	9975	5.2925
5.3.2.1	97	16175	8.085833
5.3.2.2	77	12271	6.247967
5.4	57	21754	8.4262
5.5.1.1	53	16829	6.815367
5.5.1.2	49	14978	6.126733
5.5.2.1	116	19877	9.829767
5.5.2.2	90	20827	9.2481

Βλέπουμε ότι οι χρόνοι εμφανίζονται να είναι από 3,1 έως 11,8 ώρες. Και από την ανάλυση μας παίρνουμε ομοίως και με πριν τα εξής σημαντικά διαγράμματα:

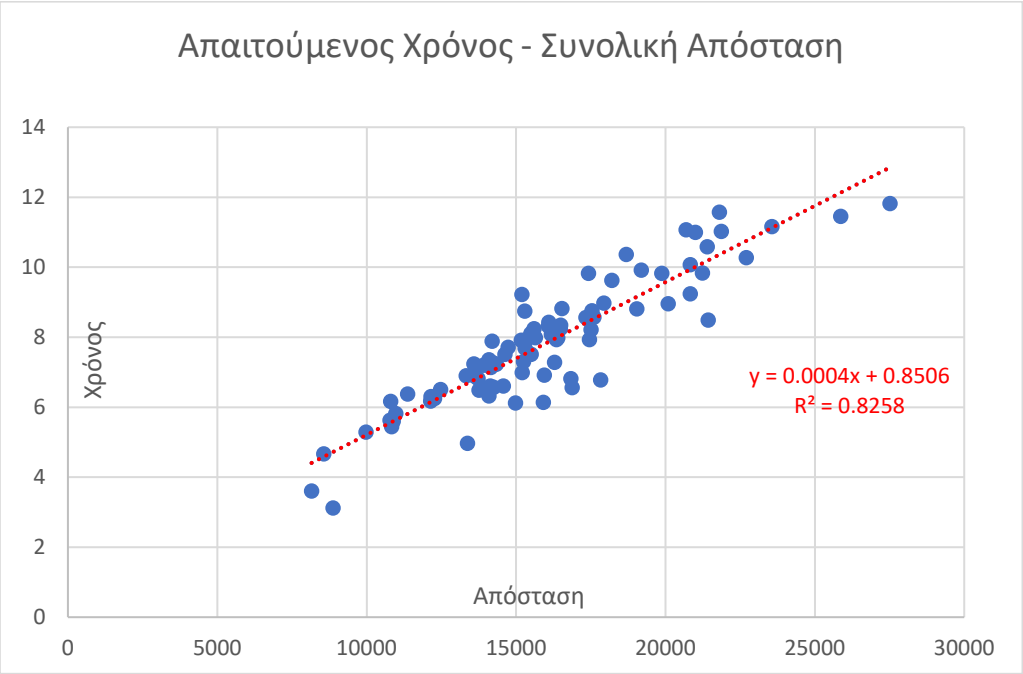


Διάγραμμα για να αποκτήσουμε μια αντίληψη πως ο αριθμός των στάσεων κάθε συστάδας επηρεάζει τον συνολικό χρόνο επιθεώρησης. Καταλληλότερη εξίσωση για μοντελοποίηση των δεδομένων κρίθηκε η y = 0.0565x + 2.6957.



Διάγραμμα για να αποκτήσουμε μια αντίληψη πως ο αριθμός των στάσεων κάθε συστάδας επηρεάζει την συνολική απόσταση της βέλτιστης διαδρομής. Καταλληλότερη εξίσωση για μοντελοποίηση των δεδομένων κρίθηκε η $y = 9832e^{0.0051x}$. Παρατηρούμε ότι από όλα τα διαγράμματα αυτό φαίνεται να είναι το πιο χαοτικό και αυτό οφείλεται στο γεγονός ότι επειδή έχουμε αρκετές συστάδες με λίγα σημεία σε σχέση με την προηγουμένη

ανάλυση η χωρική πυκνότητα των σημείων δεν είναι τόσο σταθερή οπότε και η συνολική απόσταση της βέλτιστης διαδρομής, που φαίνεται να επηρεάζεται από την χωρική πυκνότητα των σημείων, εμφανίζει μια μη προβλέψιμη διακύμανση.



Διάγραμμα για να αποκτήσουμε μια αντίληψη πως το μήκος της συνολικής απόστασης επηρεάζει τον συνολικό χρόνο επιθεώρησης. Καταλληλότερη εξίσωση για μοντελοποίηση των δεδομένων κρίθηκε η y = 0.0004x + 0.8506. Και όντως τα δεδομένα μας φαίνεται να εμφανίζουν γραμμική συμπεριφορά.

4.7 Σημαντικά αποτελέσματα

Γρήγορη επίλυση

Από κάθε διαδικασία επίλυσης που εκτελέσαμε, έχουμε μερικά αποτελέσματα που είναι πολύ κοντά στην επιθυμητή ομαδοποίηση των στάσεων που θα θέλαμε να καταφέρουμε ώστε η επιθεώρηση του δικτύου να βελτιστοποιηθεί. Θα δούμε αναλυτικά αυτά τα αποτελέσματα για να μπορέσουμε να καταλάβουμε ποια είναι τα στοιχεία που κάνουν μια ομαδοποίηση επιθυμητή. Θα δούμε όλα τα αποτελέσματα που ο τελικός χρόνος επιθεώρησης του δικτύου δεν απέχει πάνω από μισή ώρα από το επιθυμητό οκτάωρο, δηλαδή όλες τις συστάδες που ο τελικός χρόνος επιθεώρησης είναι από 7,5 έως 8,5 ώρες.

Για την γρήγορη επίλυση έχουμε στον παρακάτω πίνακα όλα τα αποτελέσματα μεταξύ των 7,5 και 8,5 ωρών:

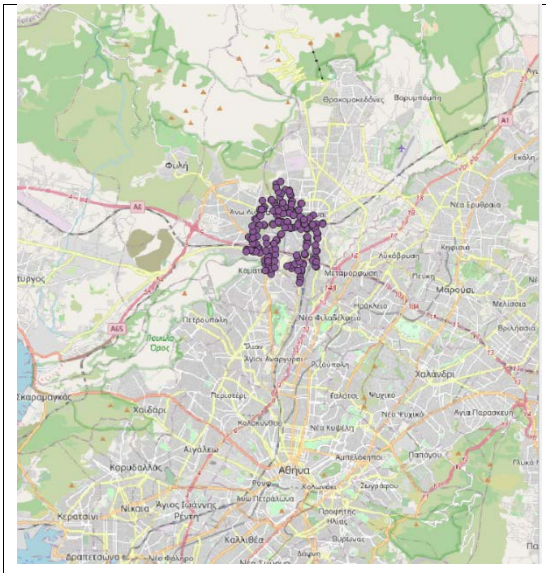
Πίνακας 4.13 Σημαντικά αποτελέσματα γρήγορης επίλυσης

Συστάδα	*Στάσεις*	*Απόσταση*	*Χρόνος*
3.1.2	189	29069	7.51035
3.2	161	33779	7.750183
3.4.2	173	32631	7.777983
3.5.1	220	31130	8.336167
3.5.3	205	30620	8.009667
4.1.4	199	28115	7.533917
4.2.2	219	28517	7.92755
4.3.2	197	28572	7.569133
4.4.2	212	32672	8.434133
5.2	136	41616	8.509067

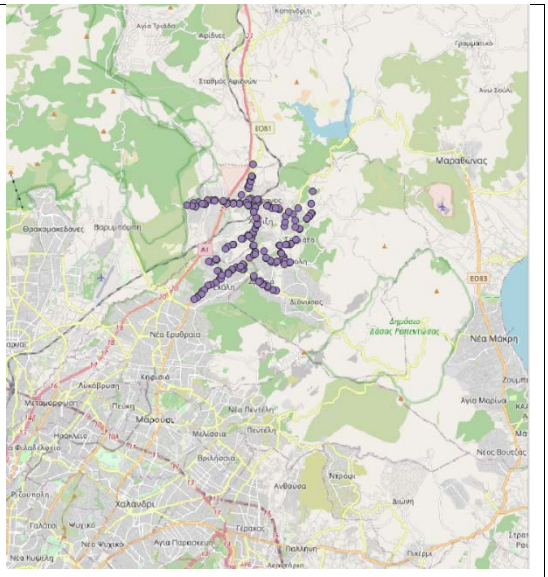
Από τα αποτελέσματα μας βλέπουμε ότι οι συστάδες που για την επιθεώρηση τους χρειάζονται κοντά στις 8 ώρες έχουν τα εξής στοιχεία:

- Ο αριθμός των στάσεων που περιέχουν είναι από 136 έως 220
- Η συνολική απόσταση που πρέπει να διανθεί είναι από 28115m έως 41616m
- Τα ακραία ζεύγη, με γνώμονα τον αριθμό των στάσεων είναι 136 στάσεις με 41616m συνολική απόσταση και 220 ζεύγη με 31130 συνολική απόσταση.
- Τα ακραία ζεύγη, με γνώμονα την απόσταση είναι 28115m συνολική απόσταση με 199 στάσεις και 41616m συνολική απόσταση με 136 στάσεις.

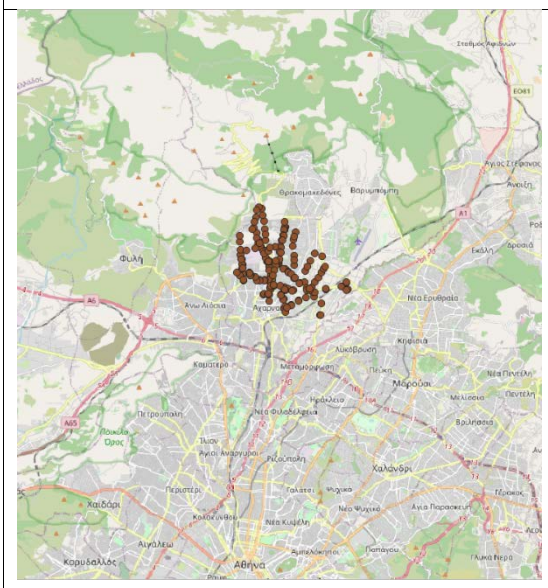
Στους παρακάτω πίνακες βλέπουμε την απεικόνιση των συστάδων αυτών στον χάρτη:



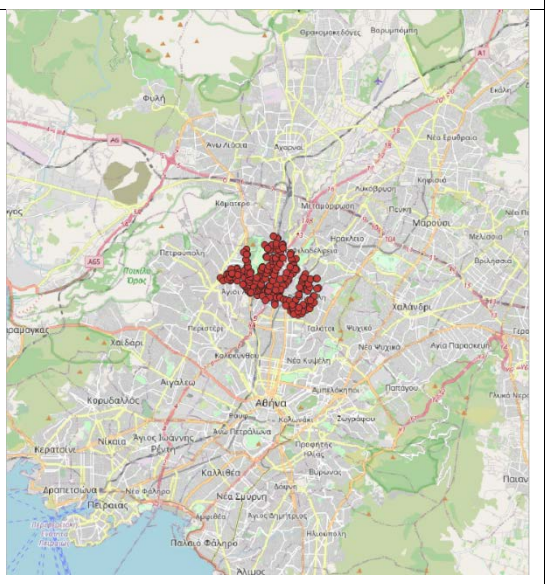
Συστάδα 3.1.2



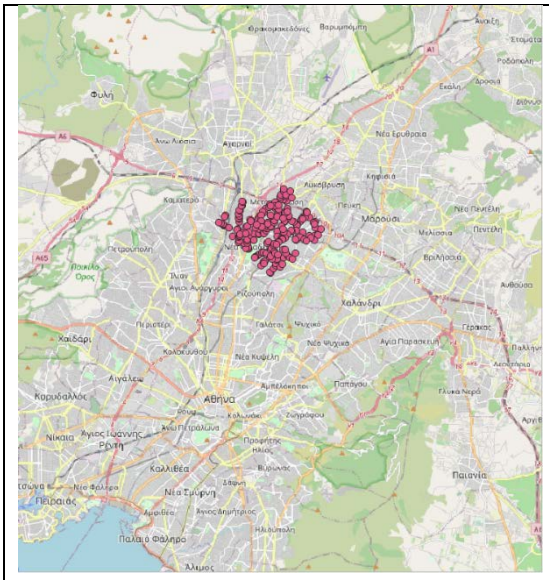
Συστάδα 3.2



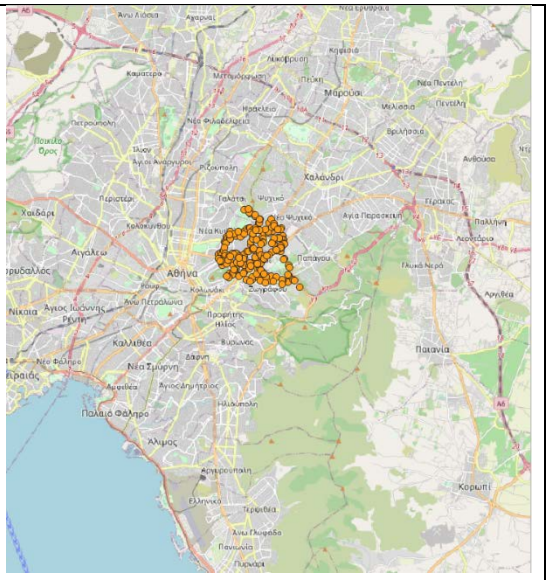
Συστάδα 3.4.2



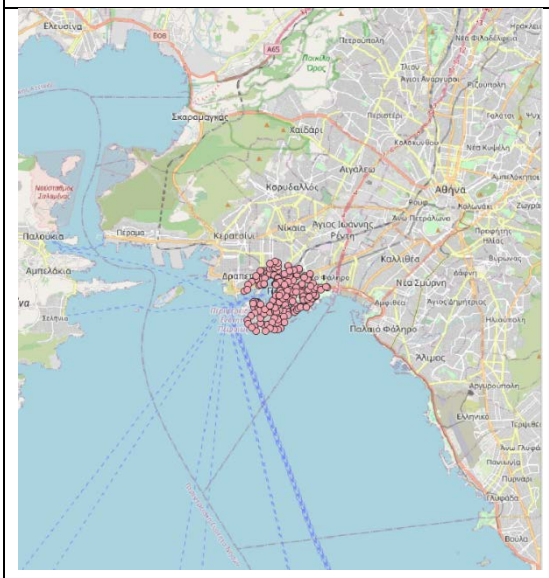
Συστάδα 3.5.1



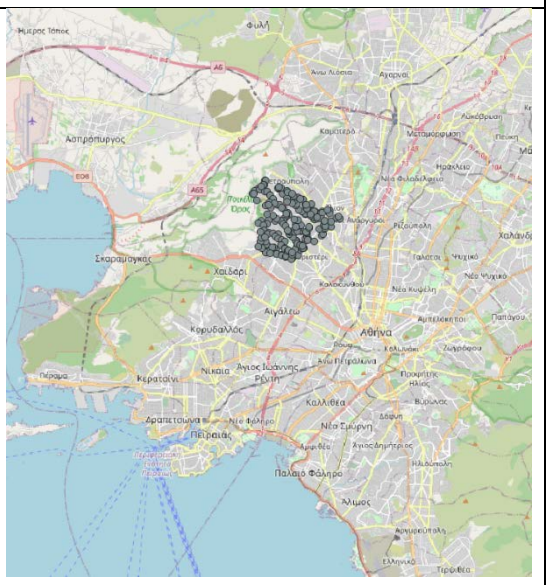
Συστάδα 3.5.3



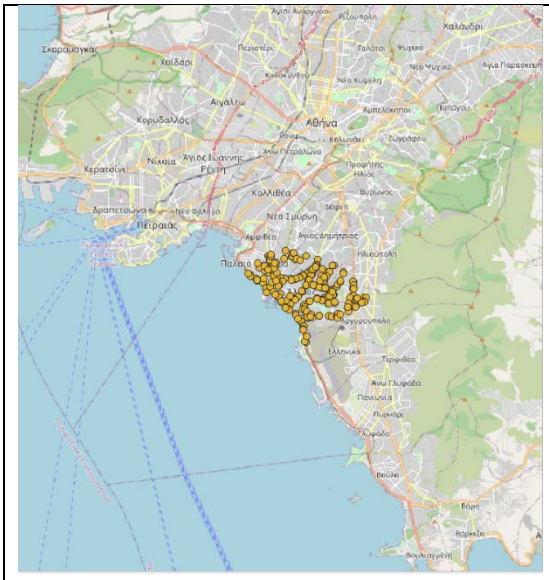
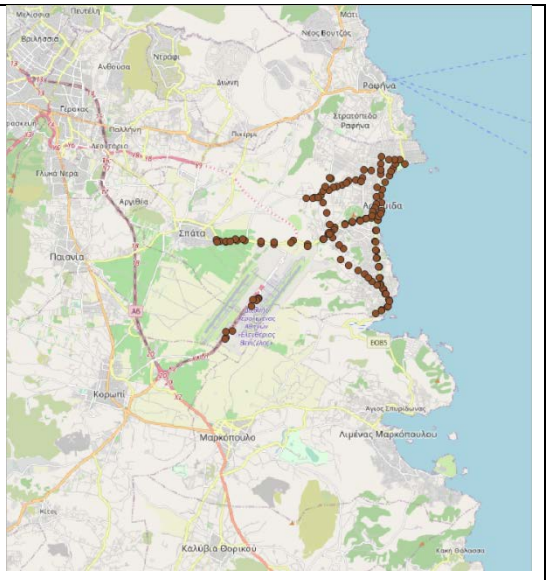
Συστάδα 4.1.4



Συστάδα 4.2.2



Συστάδα 4.3.2

	
Συστάδα 4.4.2	**Συστάδα 5.2**

Από τα αποτελέσματα μας παρατηρούμε ότι η ιδανική ομαδοποίηση του δικτύου για να ολοκληρώνεται η επιθεώρηση των συστάδων κοντά στις 8 ώρες θα ήταν τέτοια ώστε όταν η χωρική πυκνότητα των στάσεων είναι μικρή να μην έχουμε πολλές στάσεις, περίπου 140, και όταν η χωρική πυκνότητα των στάσεων είναι μεγάλη να έχουμε πιο πολλές στάσεις περίπου 210.

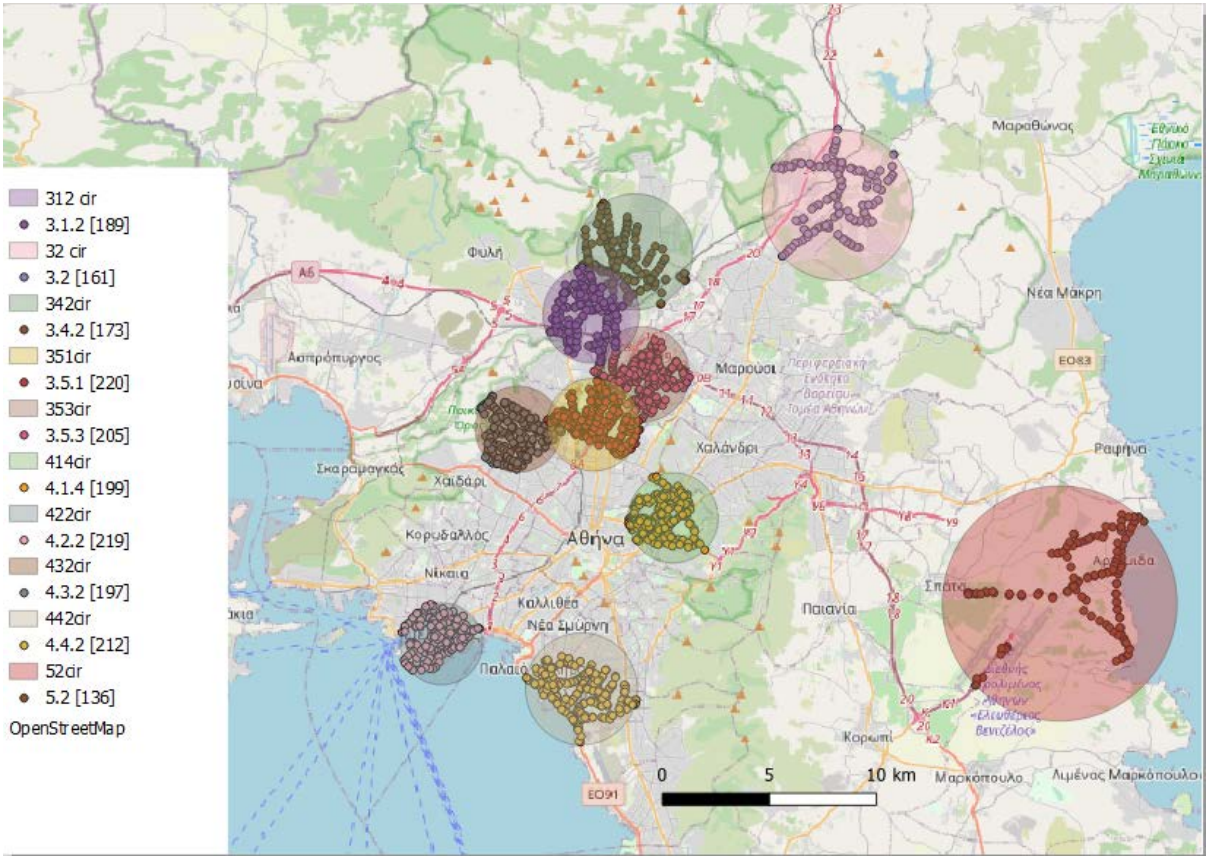
Για την ποσοτικοποίηση του ορού χωρική πυκνότητα στο πρόβλημα μας θα χρησιμοποιήσουμε την λειτουργία του λογισμικού QGIS, όπως είδαμε και στο κεφάλαιο 3.1.2, που δημιουργεί τον κύκλο ελάχιστης ακτίνας που περικλείει όλα τα σημεία, με αυτό τον τρόπο θα έχουμε καλύτερη αντίληψη ποσά σημεία θα πρέπει να περιέχει η συστάδα μας.

Στον παρακάτω πίνακα φαίνεται η αντιστοίχιση σε κάθε συστάδα μας της ελάχιστης διαμέτρου του κύκλου που περικλείει όλα τα σημεία.

Πίνακας 4.14 Ελάχιστη γεωμετρία σημαντικών αποτελεσμάτων γρήγορης επίλυσης

Συστάδα	*Στάσεις*	*Απόσταση*	*Χρόνος*	*Διάμετρος(m)*	*Εμβαδόν(km²)*
3.1.2	189	29069	7.51035	2,254	16
3.2	161	33779	7.750183	3521.6	39
3.4.2	173	32631	7.777983	2708.9	23.1
3.5.1	220	31130	8.336167	2160.4	14.7
3.5.3	205	30620	8.009667	2227.1	15.6
4.1.4	199	28115	7.533917	2102.4	13.9
4.2.2	219	28517	7.92755	1918	11.6
4.3.2	197	28572	7.569133	2012.9	12.8
4.4.2	212	32672	8.434133	2614	21.5
5.2	136	41616	8.509067	5487	94.6

Και εδώ έχουμε τον χάρτη όπου φαίνεται η απεικόνιση των κύκλων που περικλείουν την κάθε συστάδα μας:



Σχήμα 4.16 Σημαντικές συστάδες και απεικόνιση της ελάχιστης γεωμετρίας

Σημαντικά αποτελέσματα – αργή επίλυση

Για την αργή επίλυση ομοίως έχουμε μερικά αποτελέσματα τα οποία είναι ιδιαίτερα επιθυμητά. Από την ομαδοποίηση των στάσεων μας για την αργή επίλυση επιθεώρησης του δικτύου παρατηρούμε ότι υπάρχουν αρκετές συστάδες για τις οποίες ο χρόνος που απαιτείται για την ολοκλήρωση της επιθεώρησης τους είναι πολύ κοντά στις 8 ώρες, είναι από 7,5 έως 8,5 ώρες. Στον παρακάτω πίνακα βλέπουμε το σύνολο των συστάδων αυτών:

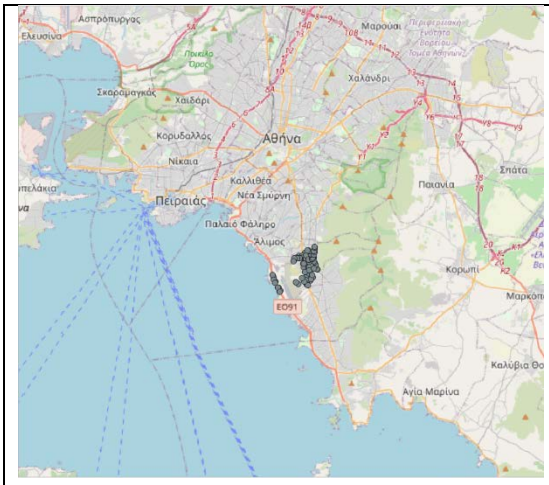
Πίνακας 4.15 Σημαντικά αποτελέσματα αργής επίλυσης

Συστάδα	*Στάσεις*	*Απόσταση*	*Χρόνος*
2.1.1.1	86	15501	7.516967
2.1.2.2	94	14620	7.519333
2.3.1	62	21426	8.494467
2.5.2	92	16393	7.984567
3.1.1.2	93	15302	7.6906
3.1.2.2	107	15603	8.247567
3.3.1.1	89	17504	8.217867
3.3.2.1	91	16344	7.936533
3.4.1.2	81	17451	7.9353
3.5.3.1	105	16080	8.324
3.5.3.2	100	15342	7.935933
4.1.1.2	99	16479	8.2437
4.1.4.2	109	14195	7.891833
4.2.4.2	108	16092	8.4276
4.3.3.1	99	14735	7.7205
5.3.1.1	104	15488	8.113067
5.3.2.1	97	16175	8.085833

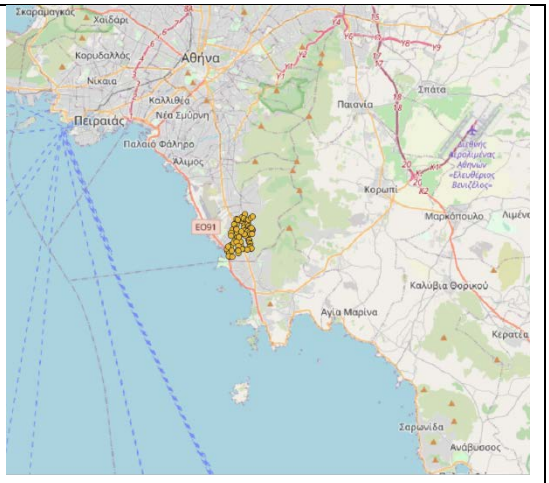
Από τα αποτελέσματα μας βλέπουμε ότι οι συστάδες που για την επιθεώρηση τους χρειάζονται κοντά στις 8 ώρες έχουν τα εξής στοιχεία:

- Ο αριθμός των στάσεων που περιέχουν είναι από 62 έως 109
- Η συνολική απόσταση που πρέπει να διανθεί είναι από 14195m έως 21426m
- Τα ακραία ζεύγη, με γνώμονα τον αριθμό των στάσεων είναι 62 στάσεις με 21426m συνολική απόσταση και 109 ζεύγη με 14195m συνολική απόσταση.
- Τα ακραία ζεύγη, με γνώμονα την απόσταση είναι 14195m συνολική απόσταση με 109 στάσεις και 21426m συνολική απόσταση με 62 στάσεις, όμοια με τα προηγούμενα.

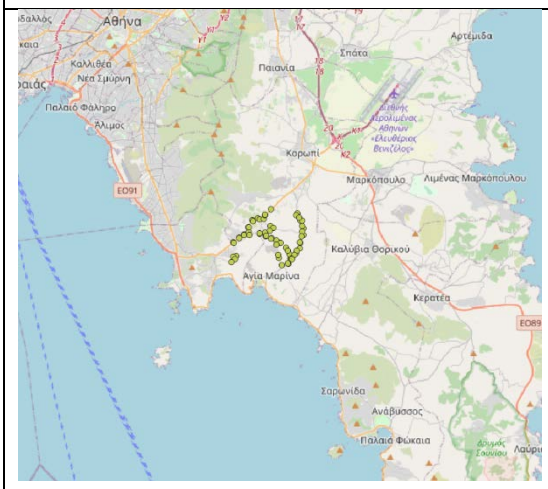
Στους παρακάτω πίνακες βλέπουμε την απεικόνιση των συστάδων αυτών στον χάρτη:



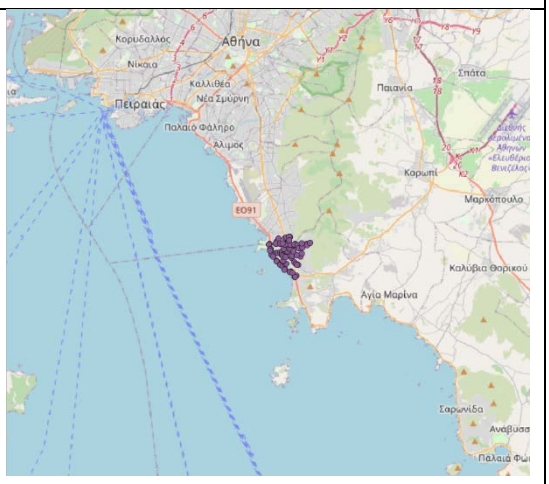
Συστάδα 2.1.1.1



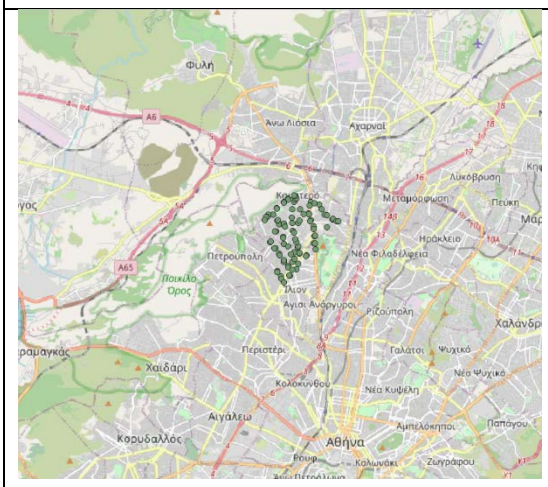
Συστάδα 2.1.2.2



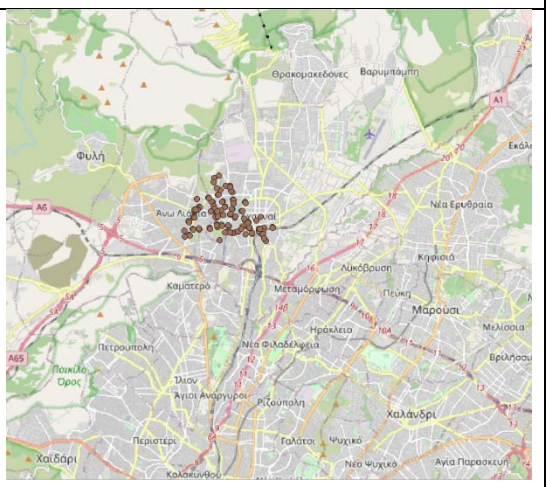
Συστάδα 2.3.1



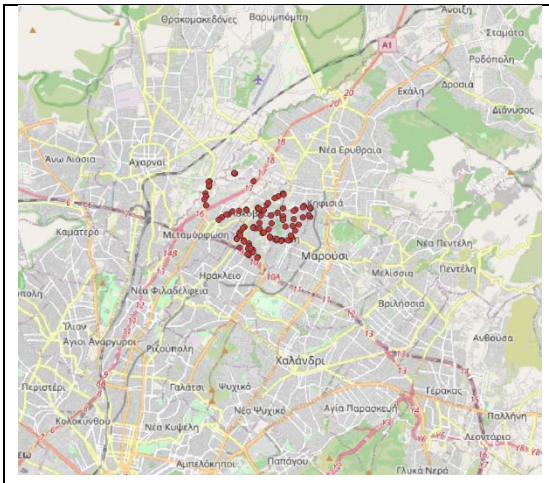
Συστάδα 2.5.2



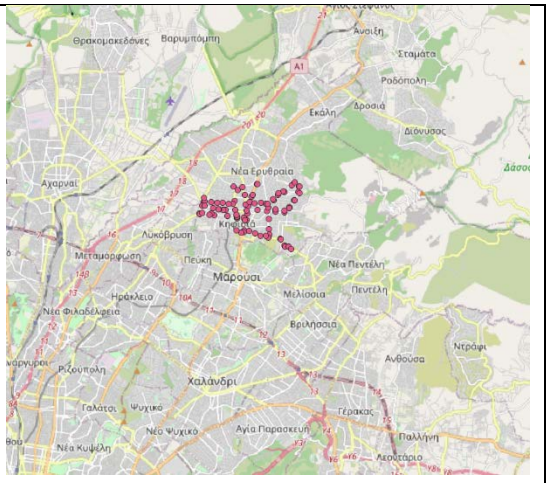
Συστάδα 3.1.1.2



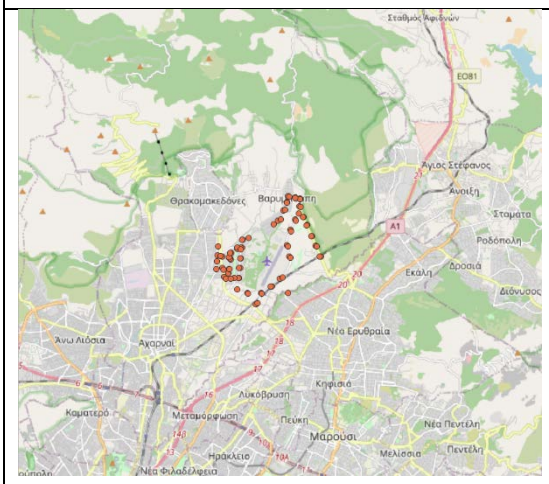
Συστάδα 3.1.2.2



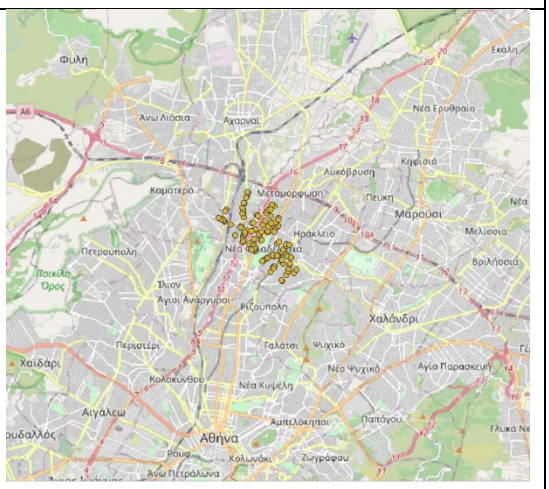
Συστάδα 3.3.1.1



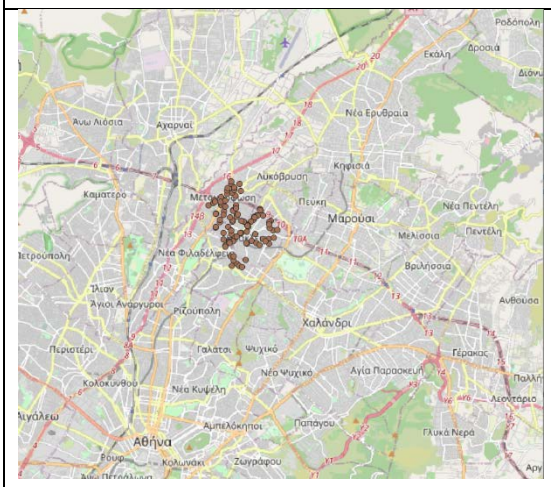
Συστάδα 3.3.2.1



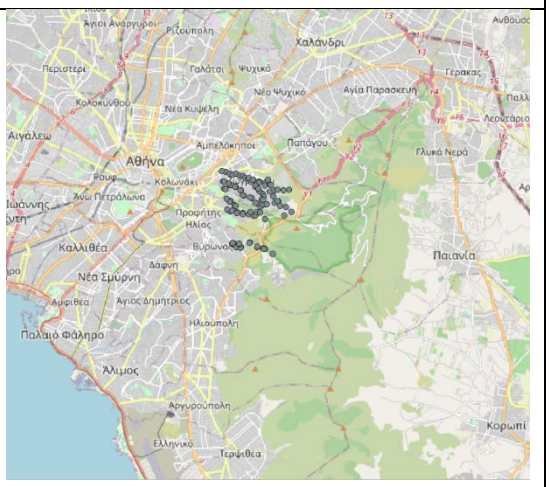
Συστάδα 3.4.1.2



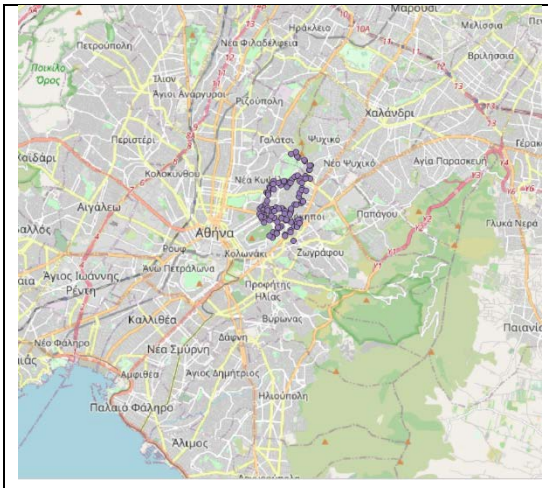
Συστάδα 3.5.3.1



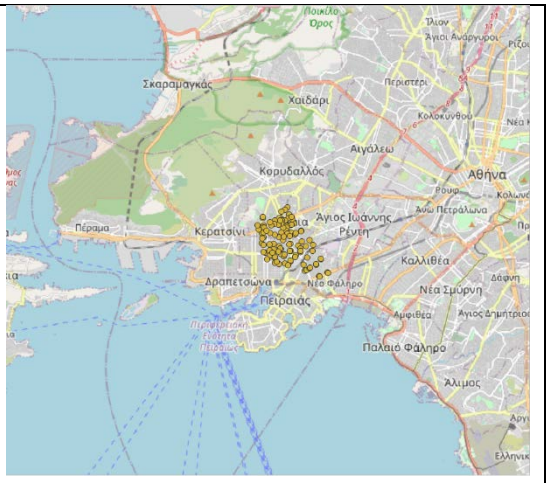
Συστάδα 3.5.3.2



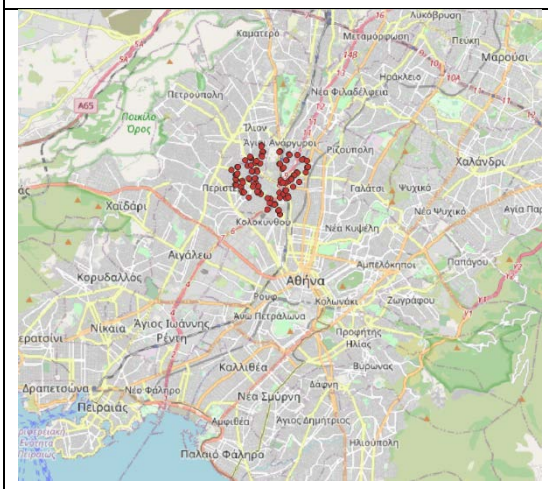
Συστάδα 4.1.1.2



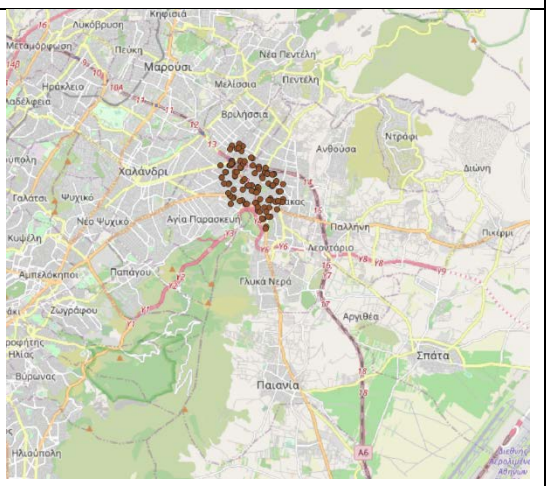
Συστάδα 4.1.4.2



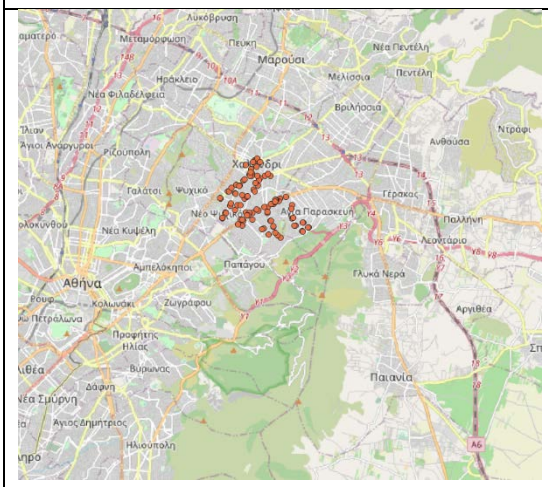
Συστάδα 4.2.4.2



Συστάδα 4.3.3.1



Συστάδα 5.3.1.1



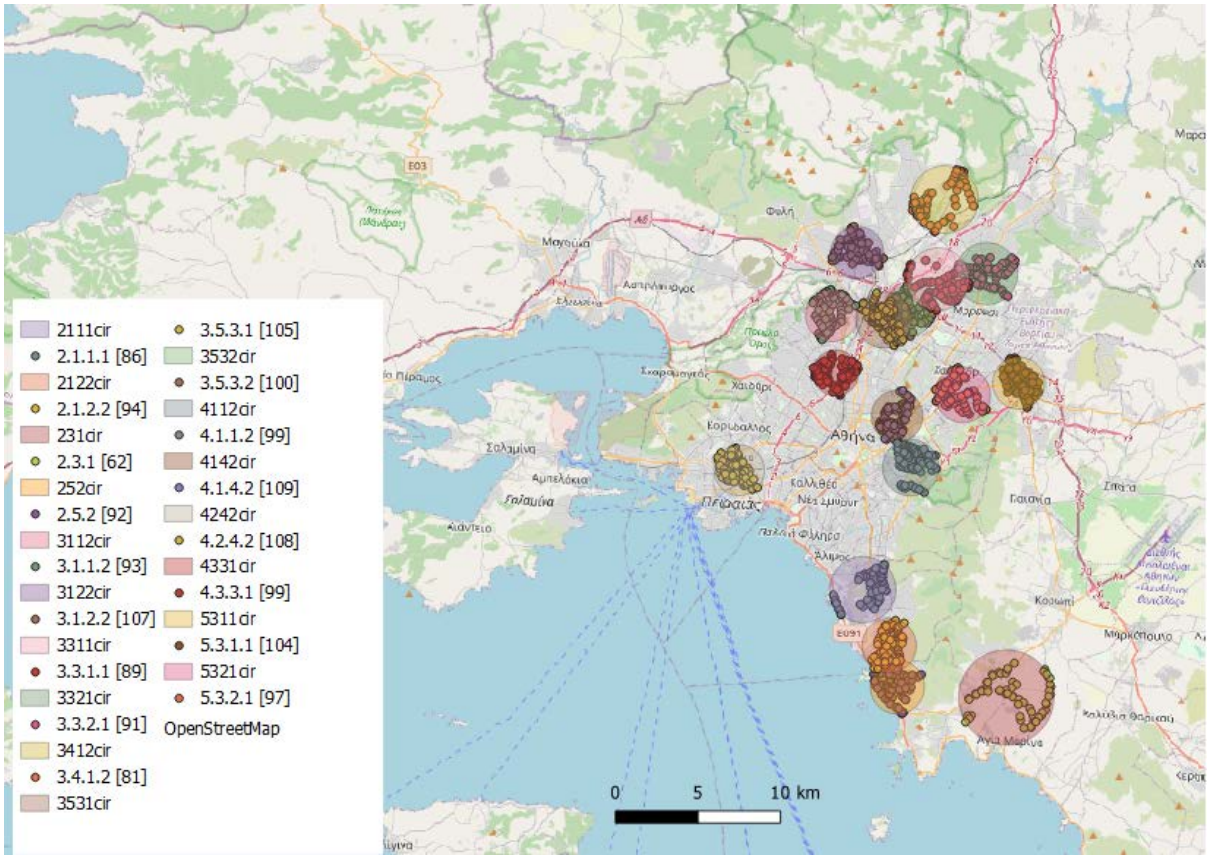
Συστάδα 5.3.2.1

Όπως είδαμε και πριν ενδιαφέρον μέγεθος για να καταλάβουμε ανάλογα την πυκνότητα των στάσεων στον χάρτη πόσες να διαλέξουμε ώστε η επιθεώρηση να μπορεί να ολοκληρωθεί κοντά στις 8 ώρες είναι η χωρική πυκνότητα των στάσεων που έχει η κάθε συστάδα. Για την ποσοτικοποίηση της θα χρησιμοποιήσουμε όπως και πριν την εύρεση του κύκλου με ελάχιστη διάμετρο που περικλείει όλες τις στάσεις. Στον παρακάτω πίνακα έχουμε τα στοιχεία των συστάδων μας και του ελάχιστου κύκλου που τις περικλείει:

Πίνακας 4.15 Σημαντικά αποτελέσματα αργής επίλυσης και ελάχιστη γεωμετρία

Συστάδα	*Στάσεις*	*Απόσταση*	*Διάμετρος(m)*	*Εμβαδόν(km^2)*
2.1.1.1	86	15501	2007.5	12.66078441
2.1.2.2	94	14620	1646.9	8.5208705
2.3.1	62	21426	2886.2	26.16991732
2.5.2	92	16393	1634.7	8.395095311
3.1.1.2	93	15302	1577.4	7.81687521
3.1.2.2	107	15603	1603.6	8.078702232
3.3.1.1	89	17504	1894.1	11.27081481
3.3.2.1	91	16344	1884.5	11.15685502
3.4.1.2	81	17451	2048.5	13.18321827
3.5.3.1	105	16080	1691.5	8.988630129
3.5.3.2	100	15342	1532.6	7.379163758
4.1.1.2	99	16479	1712.5	9.213203548
4.1.4.2	109	14195	1566.4	7.708233373
4.2.4.2	108	16092	1527.3	7.328215117
4.3.3.1	99	14735	1390.2	6.071612889
5.3.1.1	104	15488	1575.5	7.798055483
5.3.2.1	97	16175	1676.2	8.826757159

Και εδώ έχουμε τον χάρτη όπου φαίνεται η απεικόνιση των κύκλων που περικλείουν την κάθε συστάδα μας:



Σχήμα 4.17 Σημαντικές συστάδες και απεικόνιση της ελάχιστης γεωμετρίας

Θα χρησιμοποιήσουμε τα στοιχεία και τις ιδέες που συλλέξαμε από αυτή την ανάλυση στο επόμενο κεφάλαιο για να προσπαθήσουμε να πέτυχουμε μια ακόμα καλύτερη ομαδοποίηση του δικτύου.

5. Αντιμετώπιση προβλήματος με χρήση της ανθρώπινης διαίσθησης και την βοήθεια του αλγορίθμου K – means και της ελάχιστης περιβάλλουσας γεωμετρίας

Ιδέες και στρατηγική αντιμετώπισης

Τα αποτελέσματα της προηγουμένης ανάλυσης πάρα την προσπάθεια μας να είναι όσο το δυνατόν πιο κοντά στις 8 ώρες ευκολά παρατηρεί κάνεις ότι δεν είναι και τόσο καλά. Ένα μικρό ποσοστό των συστάδων είναι κοντά στις 8 ώρες. Επίσης ένα ακόμα πρόβλημα είναι ότι παρουσιάζονται μερικές ακραίες τιμές που δεν είναι καθόλου επιθυμητές, στην γρήγορη έχουμε ελάχιστο και μέγιστο χρόνο τις τιμές 1,7 και 11,6 ώρες αντίστοιχα και στην αργή 3,1 και 11,8. Ωστόσο από την ανάλυση των δεδομένων μας μπορούμε να συμπεράνουμε μερικά στοιχεία που μας οδηγούν στο ποια συστάδα έχει καλές πιθανότητες να δώσει τελικό χρόνο επιθεώρησης κοντά στις 8 ώρες.

Για την γρήγορη επίλυση όπου η χωρική πυκνότητα είναι μεγάλη δηλαδή όλα τα σημεία βρίσκονται εντός κύκλου ακτίνας περίπου 2 χιλιομέτρων οι συστάδες μας αν περιέχουν κοντά στις 200 στάσεις φαίνεται το τελικό αποτέλεσμα να είναι κοντά στις 8 ώρες και επίσης όπου η χωρική πυκνότητα είναι μικρή δηλαδή τα σημεία να βρίσκονται εντός κύκλου ακτίνας περίπου 5 χιλιομέτρων αν περιέχουν κοντά στις 130 στάσεις.

Για την αργή επίλυση όπου η χωρική πυκνότητα είναι μεγάλη δηλαδή όλα τα σημεία βρίσκονται εντός κύκλου ακτίνας περίπου 1,4 χιλιομέτρων οι συστάδες μας αν περιέχουν κοντά στις 100 στάσεις φαίνεται το τελικό αποτέλεσμα να είναι κοντά στις 8 ώρες και επίσης όπου η χωρική πυκνότητα είναι μικρή δηλαδή τα σημεία να βρίσκονται εντός κύκλου ακτίνας περίπου 3 χιλιομέτρων αν περιέχουν κοντά στις 60 στάσεις.

Ο λόγος που ο αλγόριθμος k – means δεν προκαλεί μια ομοιόμορφη συσταδιοποιήση ως προς τον χρόνο είναι αρχικά γιατί η αρχιτεκτονική του δεν είναι χτισμένη γύρω από κάποια τέτοια παράμετρο πάρα μόνο τα σημεία να ανήκουν στην συστάδα από της οποίας το κέντρο απέχουν την ελάχιστη απόσταση. Το γεγονός αυτό λόγω μη ομοιόμορφης τοποθέτησης στον γεωγραφικό χώρο των στάσεων άλλοτε προκαλεί την δημιουργία συστάδων με μικρό αριθμό στάσεων και άλλοτε συστάδες με μεγάλο αριθμό στάσεων.

5.1 Χρήση αλγορίθμων και δημιουργία υπερσυστάδων κατά επιλογήν

Προκειμένου να δημιουργήσουμε συστάδες που να έχουμε καλύτερη πιθανότητα να μας δώσουν καλά αποτελέσματα με τα εργαλεία που έχουμε στα χεριά μας θα εργαστούμε ως εξής:

- Θα υποδιαιρέσουμε το δίκτυο σε έναν μεγάλο αριθμό συστάδων τουλάχιστον 5 φορές μεγαλύτερο από το σύνολο των τελικών συστάδων που περιμένουμε να προκύψουν.
- Για κάθε μια από τις συστάδες θα βρούμε τον ελάχιστο κύκλο που τις περικλείει
- Θα ενώσουμε συστάδες για την δημιουργία μεγαλύτερων συστάδων που θα έχουν αρκετά καλή πιθανότητα να δίνουν χρόνο κοντά στις 8 ώρες σύμφωνα με τα στοιχεία που είδαμε.

Μέχρι στιγμής η ανάλυση μας έχει δώσει τα εξής στοιχεία:

Πίνακας 5.1 Χρονικά Συμπεράσματα για αργή και γρήγορη επίλυση

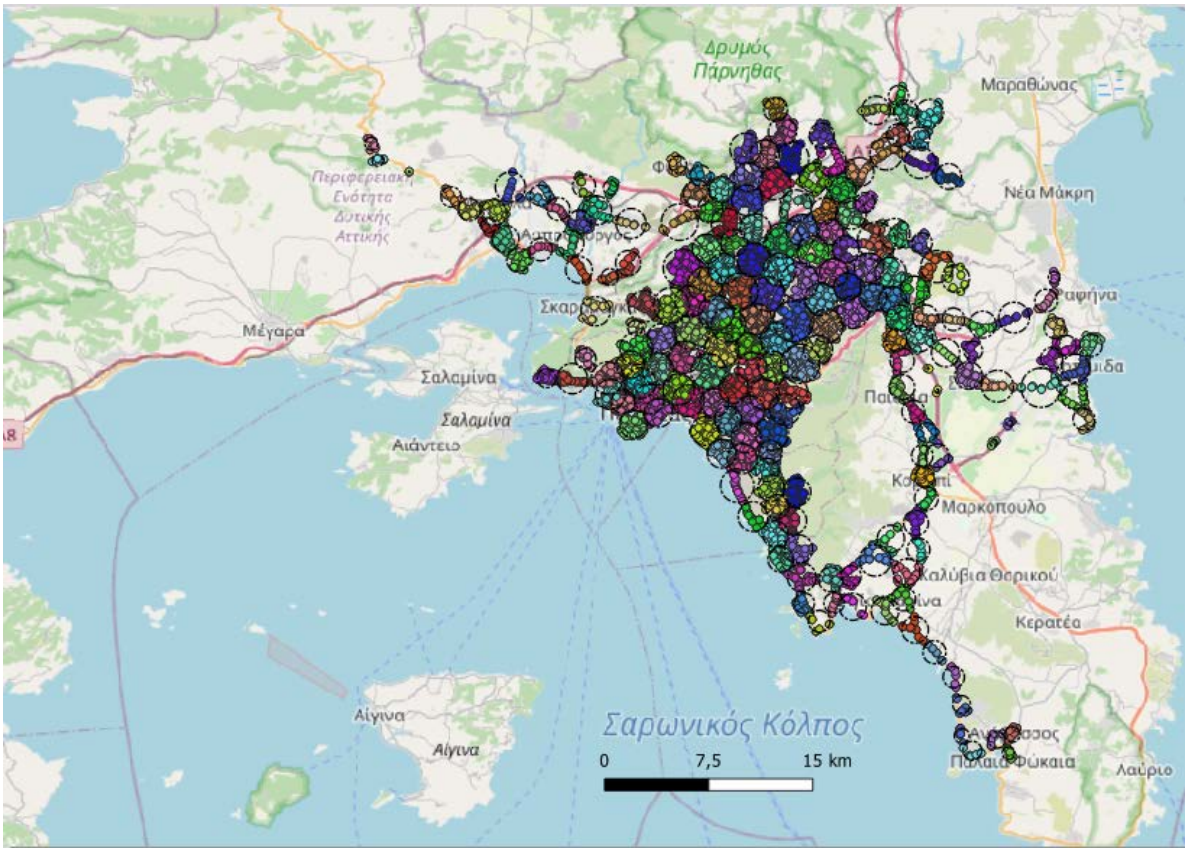
Επίλυση	*Σύνολο Συστάδων*	*Συνολικός χρόνος*	*Μέσος χρόνος ανά συστάδα*
Γρήγορη	44	342.14965	7.776128409
Αργή	89	697.8107	7.840569663

Βλέπουμε ότι κατά μέσο όρο ο χρόνος που έχουμε είναι αρκετά κοντά στις 8 ώρες.

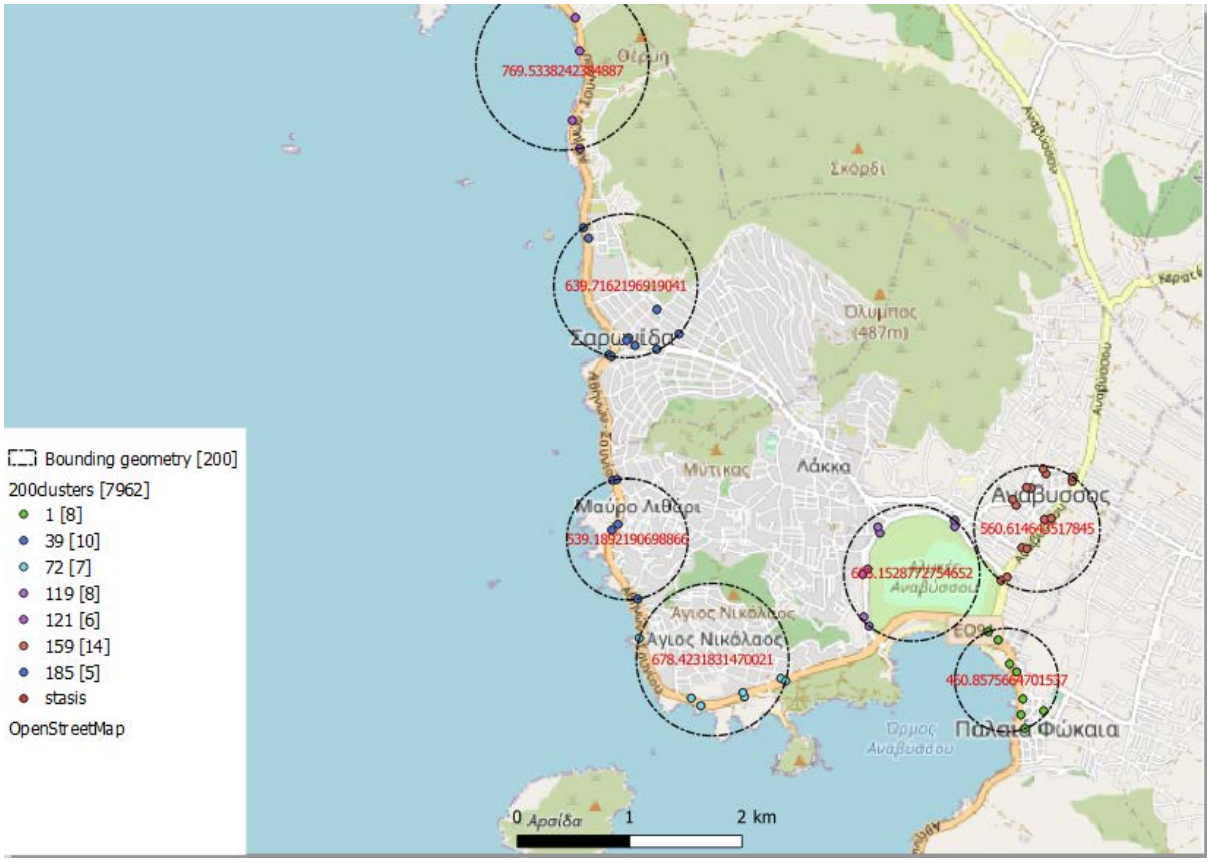
Οπότε θα διαλέξουμε να υποδιαιρέσουμε το δίκτυο σε 200 συστάδες για την γρήγορη επίλυση και έπειτα σε 500 συστάδες για την αργή.

Γρήγορη Επίλυση

Αφού τεμαχίσουμε το συνολικό δίκτυο στάσεων σε 200 κομμάτια και βρούμε τον κύκλο ελάχιστης ακτίνας που τα περικλείει θα έχουμε τον εξής χάρτη στον οποίο θα μπορούμε να εργαστούμε.

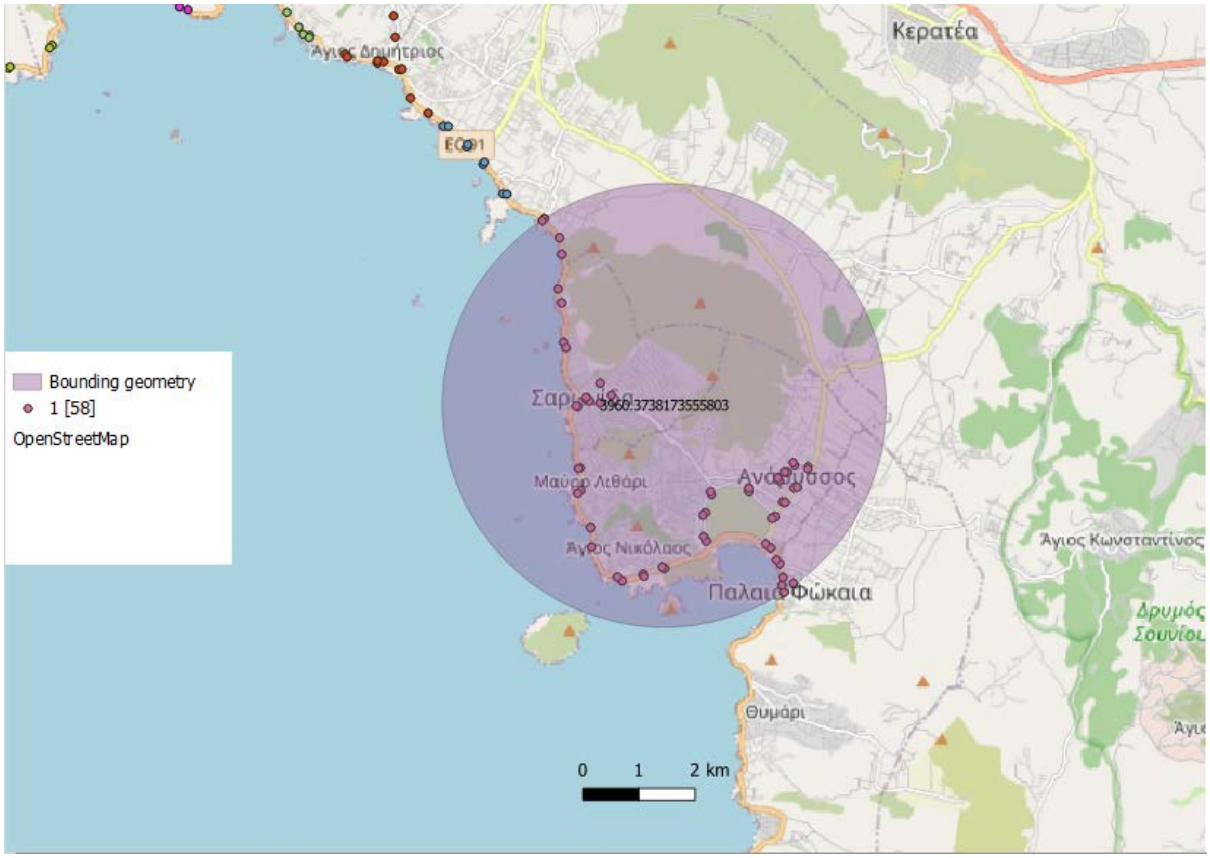


Και άμα μεγεθύνουμε σε μια περιοχή για να δημιουργήσουμε την πρώτη μας υπερσυστάδα βλέπουμε:



Όπως καταλαβαίνουμε από την ένωση αυτών των συστάδων θα πρόκυπτε μια υπερσυστάδα που θα είχε 8 + 10 +7 +8 + 6 + 14 + 5 = 58 στοιχεία και χρησιμοποιώντας και το εργαλείο measure line του QGIS θα μπορούσαμε να δούμε ότι ο κύκλος που θα την περιέκλειε θα ήταν κοντά στα 4 χιλιόμετρα (αποκτάμε μια αντίληψη για την ακτίνα του κύκλου και χωρίς την λειτουργία measure line λόγω των ακτινών των κύκλων που αναγράφονται στις συστάδες).

Τελικά θα πρόκυπτε η πρώτη μας υπερσυστάδα:

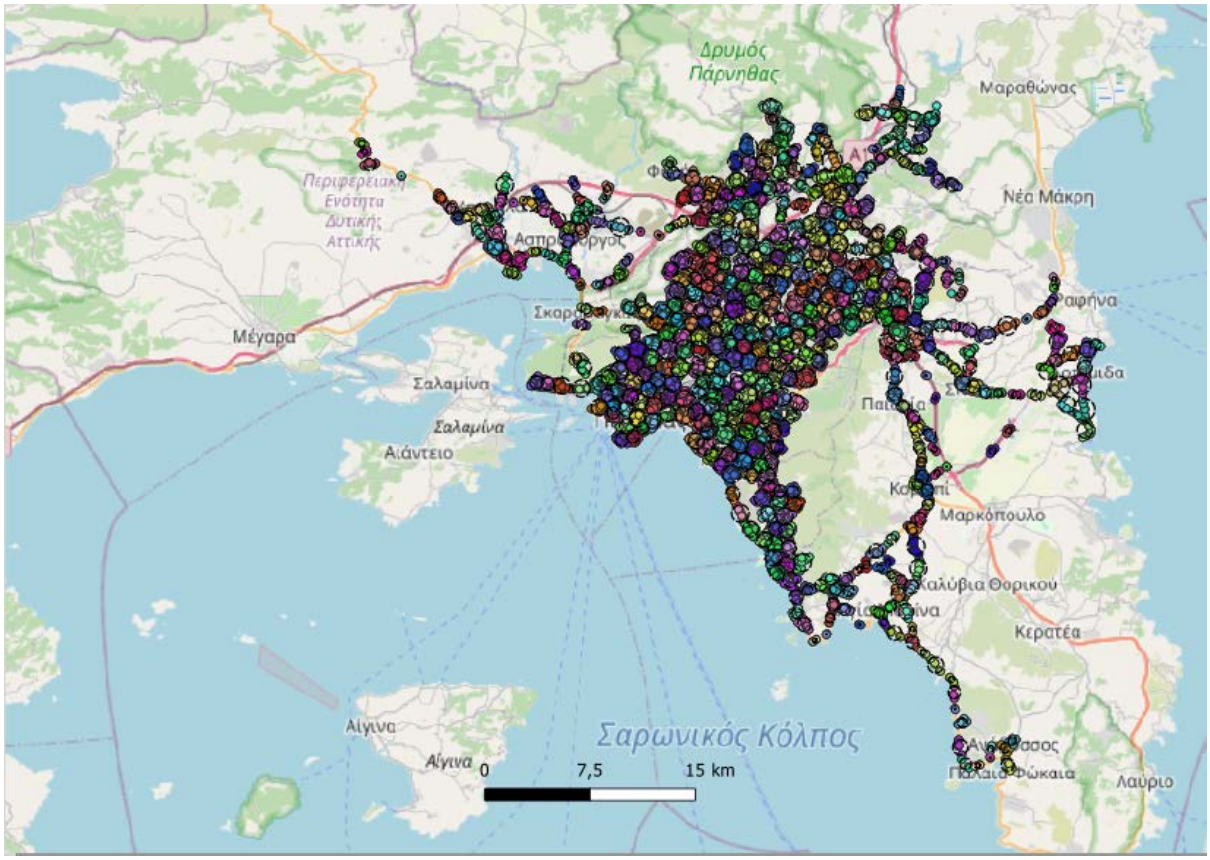


<u>Σχήμα 5.1 δημιουργία υπερσυστάδας</u>

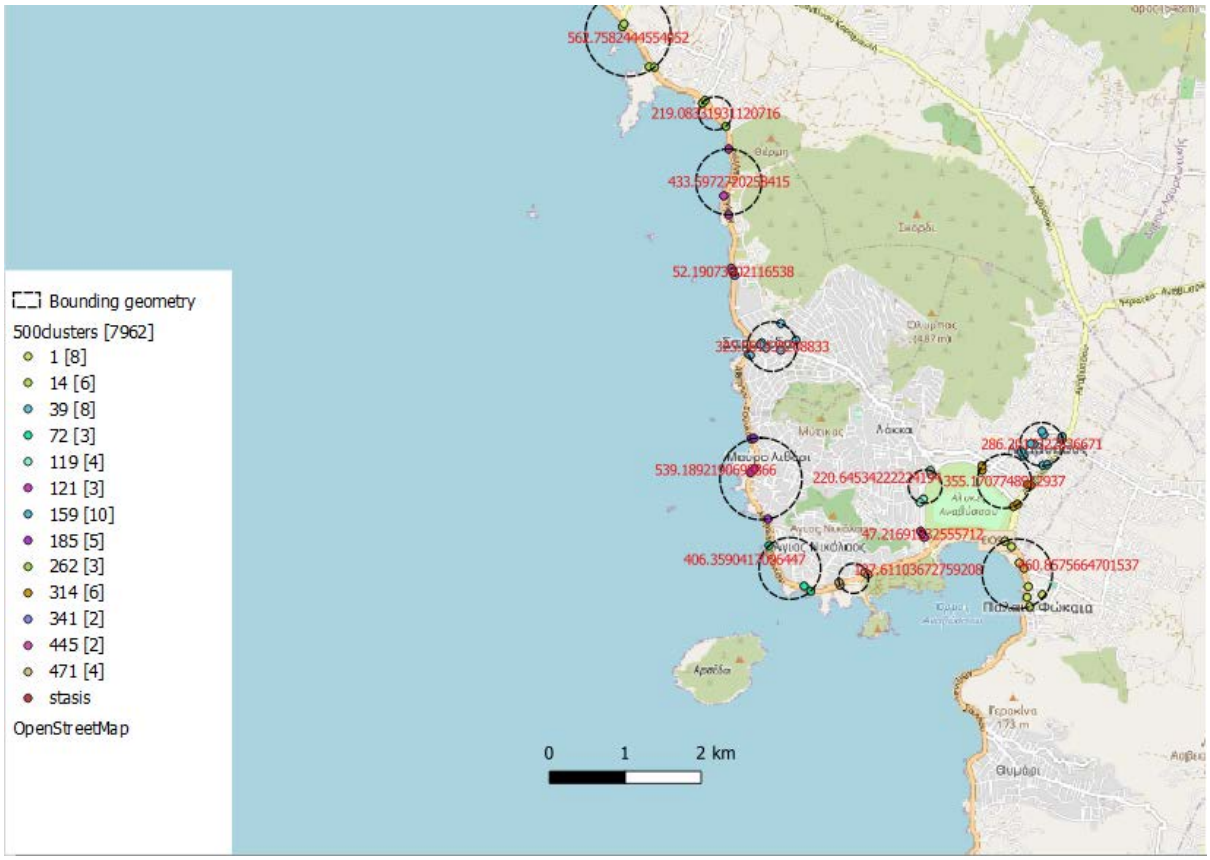
Η οποία όπως βλέπουμε περιέχει 58 στάσεις εντός 3,96 χιλιομέτρων.

Αργή Επίλυση

Όμοια για την επίτευξη κάλων υπερσυστάδων κατά την αργή επίλυση θα υποδιαιρέσουμε το δίκτυο μας σε 500 συστάδες και θα εργαστούμε με όμοιο τρόπο.



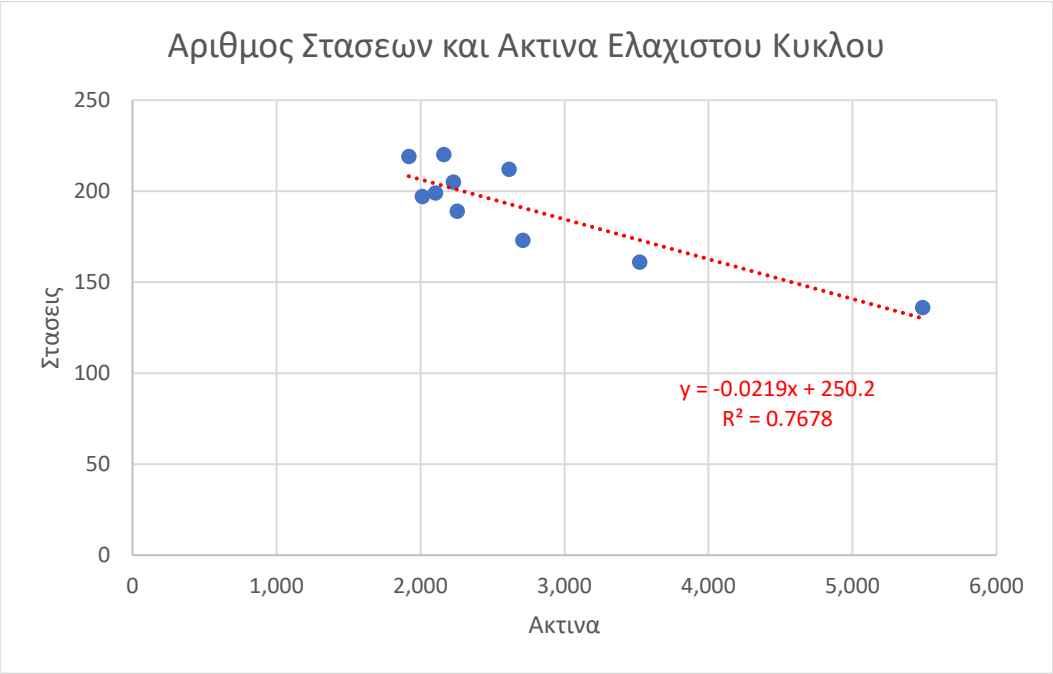
Και αν μεγεθύνουμε στην ιδιά περιοχή που δουλέψαμε την πρώτη μας υπερσυστάδα για την γρήγορη επίλυση βλέπουμε:



Οι ίδιες 58 στάσεις απλά αυτή την φορά είναι απλωμένες σε 12 συστάδες ενώ πριν ήταν σε 7.

Για την δημιουργία των τελικών υπερσυστάδων σύμφωνα με την ανάλυση που κάναμε στο 4ο κεφάλαιο (συγκεκριμένα στο 4.7) θα προσπαθήσουμε να ακολουθήσουμε όσο πιο πολύ μπορούμε τα παρακάτω διαγράμματα.

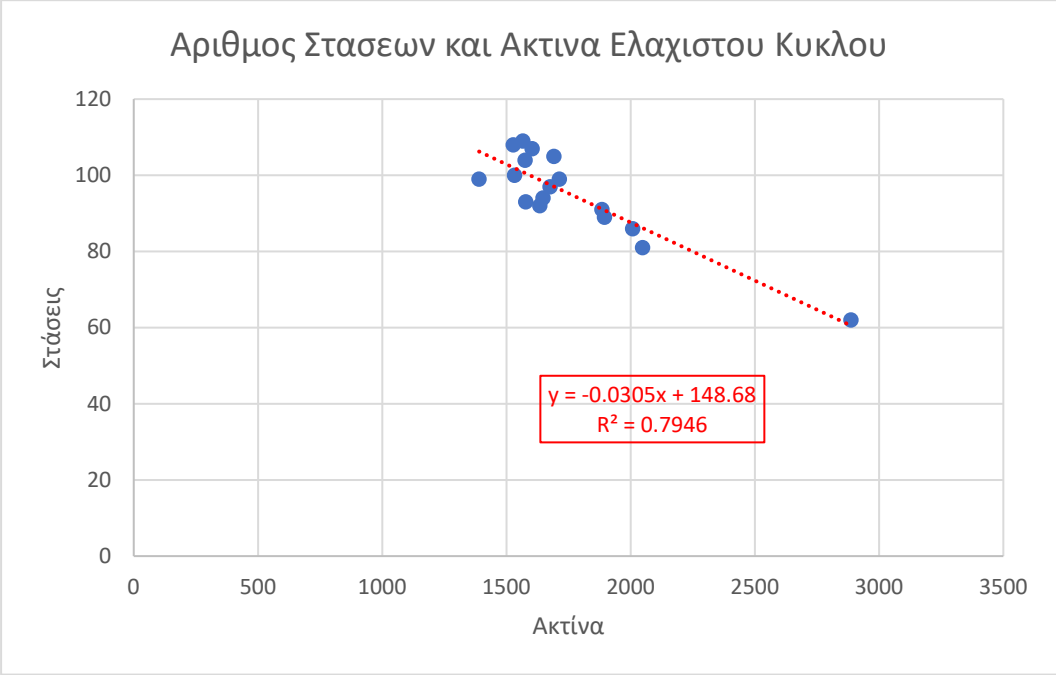
Γρήγορη επίλυση



Όπου βλέπουμε την σχέση y = -0.0219x + 250.2 που συνδέει τον αριθμό των στάσεων(μέγεθος y) με την ακτίνα του ελάχιστου περικλειόμενου κύκλου (μέγεθος x) για τα 10 καλά αποτελέσματα (7.5-8.5 ώρες) που προέκυψαν από την ανάλυση της γρήγορης επίλυσης στο κεφάλαιο 4.

Όμοια θα προσπαθήσουμε να δημιουργήσουμε υπερσυστάδες που να ακολουθούν αυτή την αναλογία.

Αργή Επίλυση



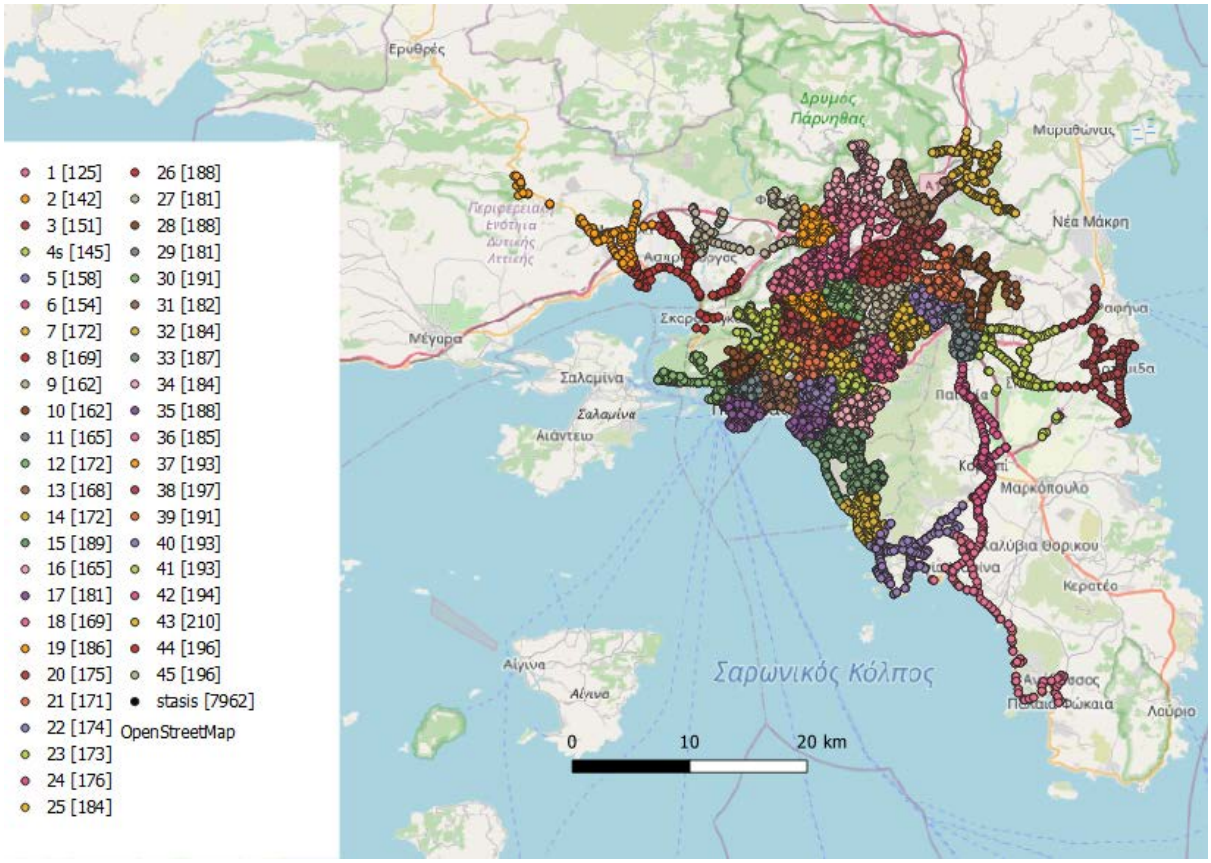
Όπου βλέπουμε την σχέση y = -0.0305x + 148.68 που συνδέει τον αριθμό των στάσεων(μέγεθος y) με την ακτίνα του ελάχιστου περικλειόμενου κύκλου (μέγεθος x) για τα 17 καλά αποτελέσματα (7.5-8.5 ώρες) που προέκυψαν από την ανάλυση της γρήγορης επίλυσης στο κεφάλαιο 4.

Όμοια θα προσπαθήσουμε να δημιουργήσουμε υπερσυστάδες που να ακολουθούν αυτή την αναλογία.

5.2 Τελική διάταξη υπερσυστάδων κατά την γρήγορη και αργή επίλυση

Με γνώμονα όσα είπαμε στις προηγούμενες σελίδες οι υπερσυστάδες που δημιουργήσαμε φαίνονται στους παρακάτω χάρτες:

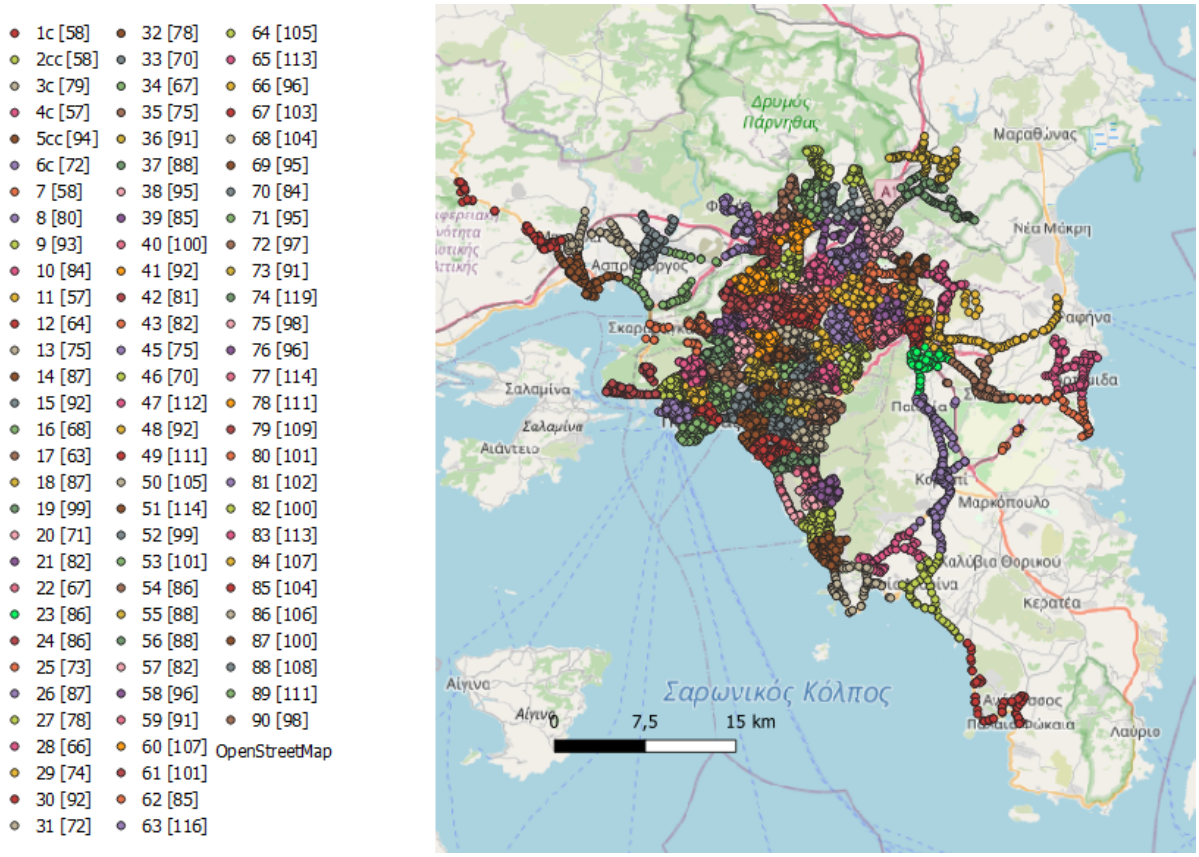
Γρήγορη επίλυση



Σχήμα 5.2 Διάταξη τελικών υπερσυστάδων γρήγορης επίλυσης

Ο χάρτης για την γρήγορη επίλυση που αποτελείται τελικά από 45 υπερσυστάδες με μέγεθος κάθε συστάδας να κυμαίνεται από 125 έως 210 στάσεις.

Αργή επίλυση



<u>Σχήμα 5.3 Διάταξη τελικών υπερσυστάδων αργής επίλυσης</u>

Ο χάρτης για την αργή επίλυση που αποτελείται τελικά από 90 υπερσυστάδες με μέγεθος κάθε συστάδας να κυμαίνεται από 57 έως 116 στάσεις.

5.3 Αποτελέσματα για κάθε υπερσυστάδα

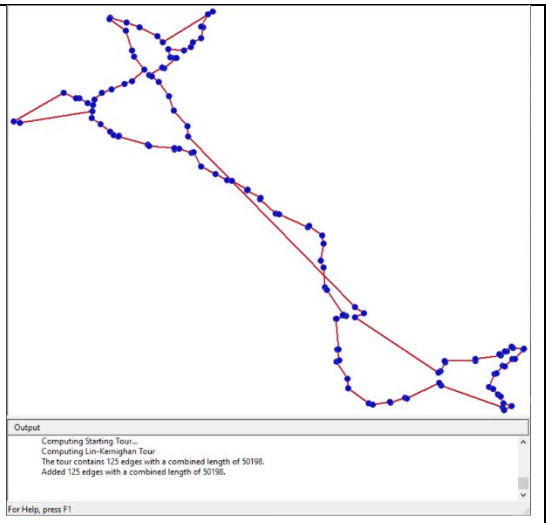
5.3.1 Αποτελέσματα Concorde TSP Solver για γρήγορη επίλυση

Αφού εξάγουμε τις συντεταγμένες για κάθε μια από τις 45 υπερσυστάδες που δημιουργήσαμε για την γρήγορη επίλυση και μορφώσουμε τα αρχεία .qs ώστε να εφαρμόσουμε τον αλγόριθμο Lin-Kernighan στο λογισμικό Concorde παίρνουμε τα εξής αποτελέσματα:

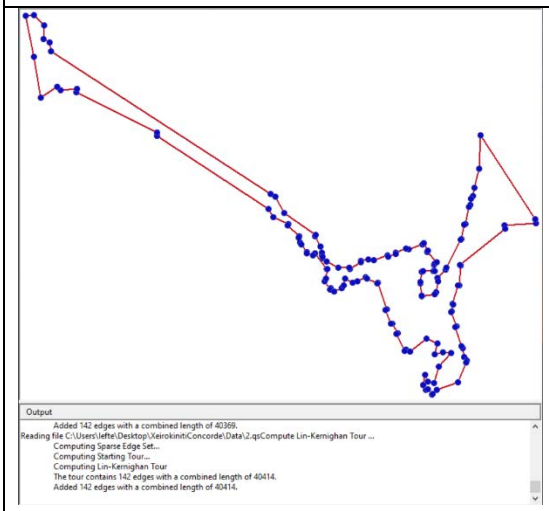
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

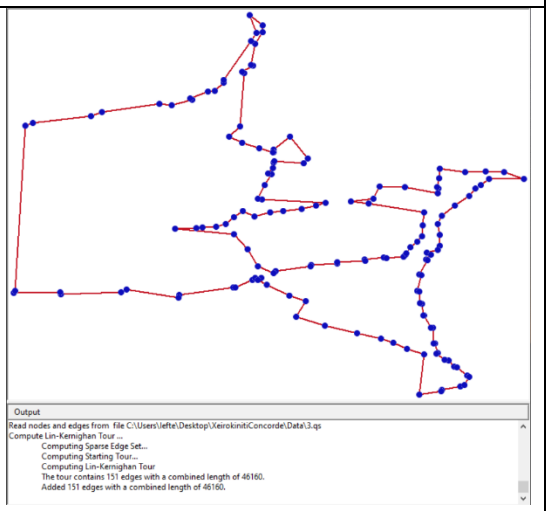
Σειρά: 1/2/3/4/5



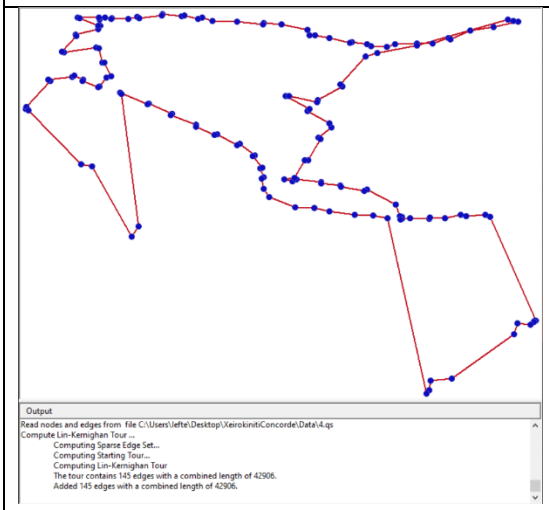
Συστάδα 1



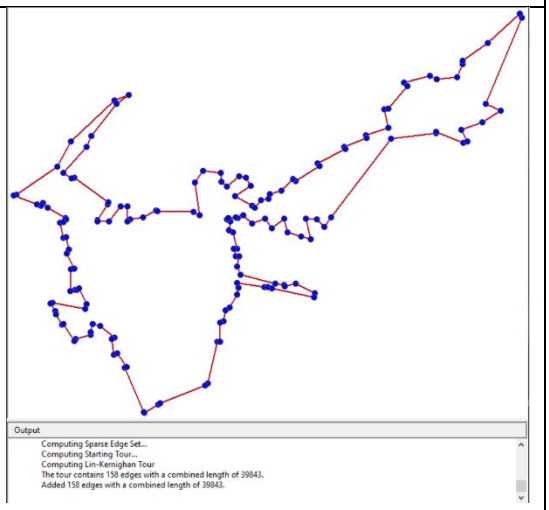
Συστάδα 2



Συστάδα 3



Συστάδα 4

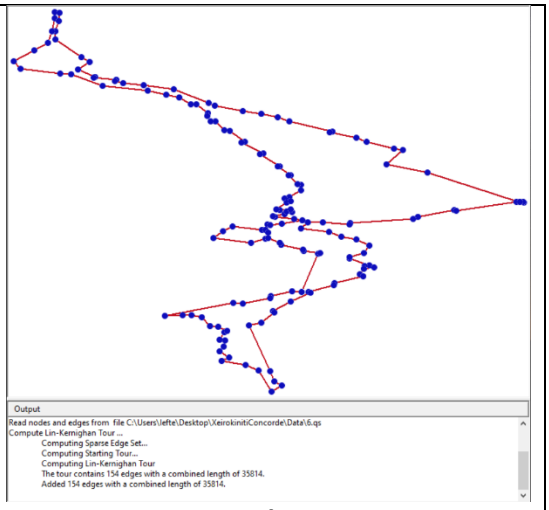


Συστάδα 5

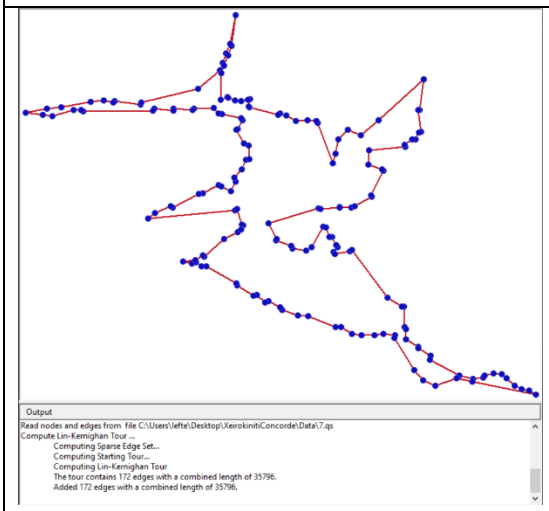
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

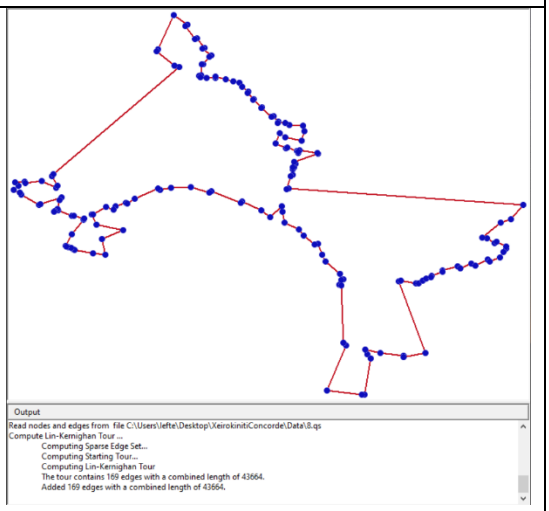
Σειρά: 6/7/8/9/10



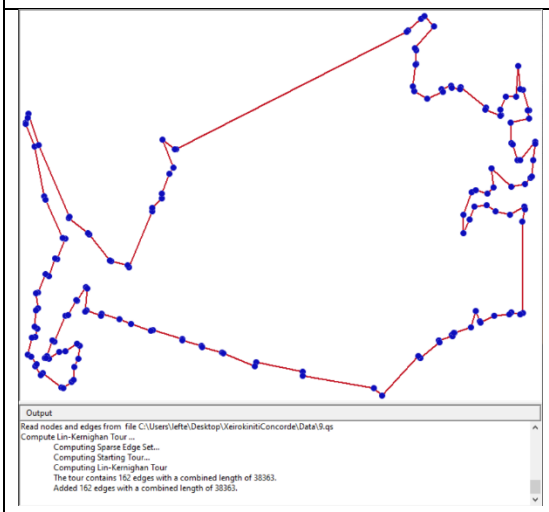
Συστάδα 6



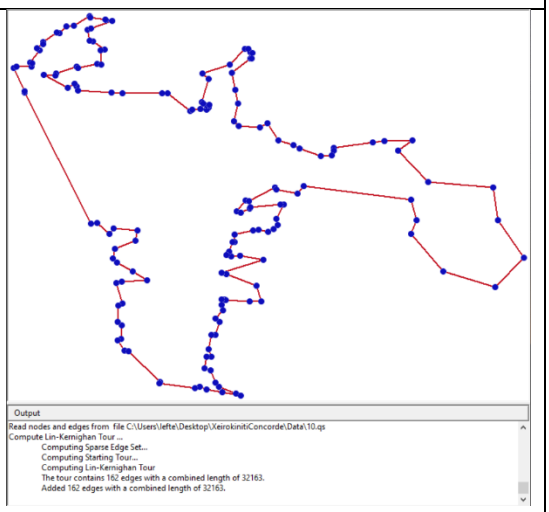
Συστάδα 7



Συστάδα 8



Συστάδα 9

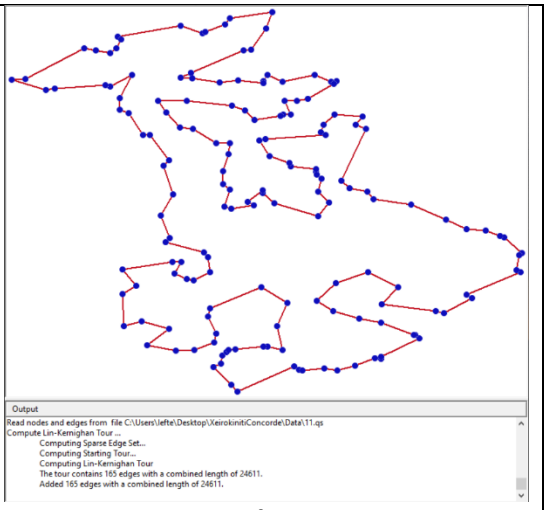


Συστάδα 10

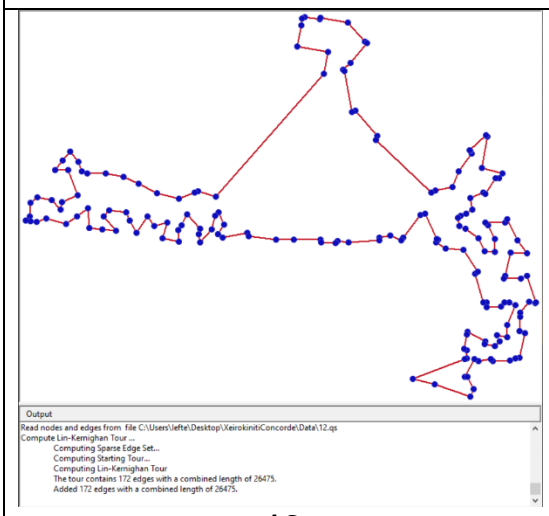
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

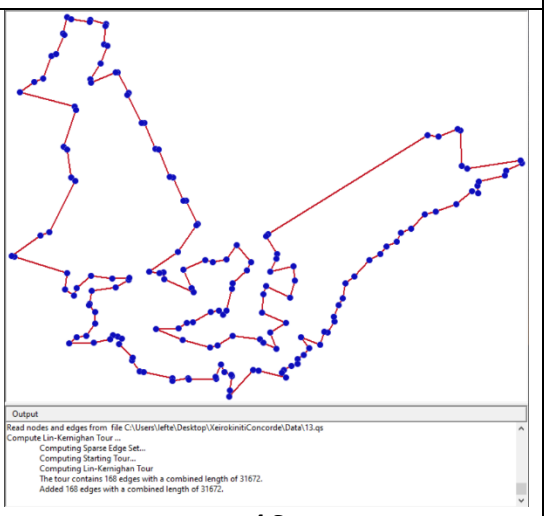
Σειρά: 11/12/13/14/15



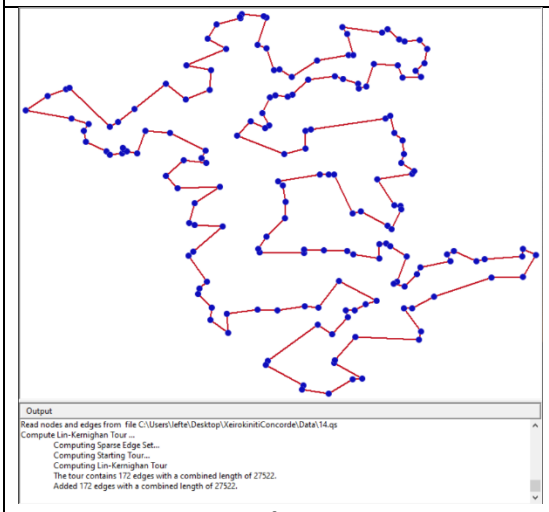
Συστάδα 11



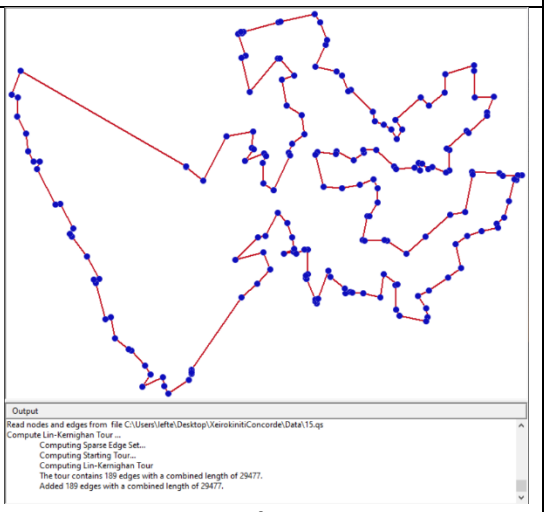
Συστάδα 12



Συστάδα 13



Συστάδα 14

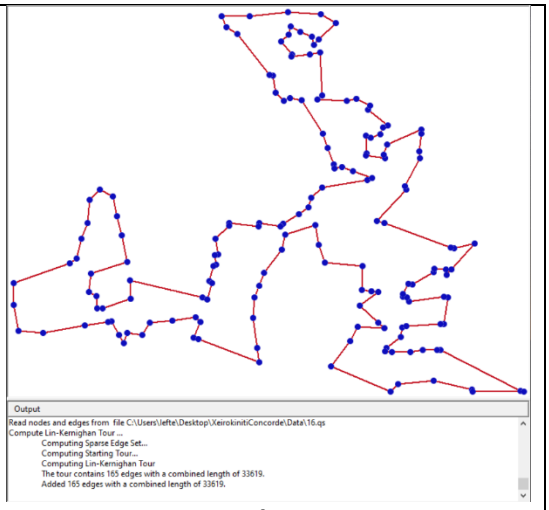


Συστάδα 15

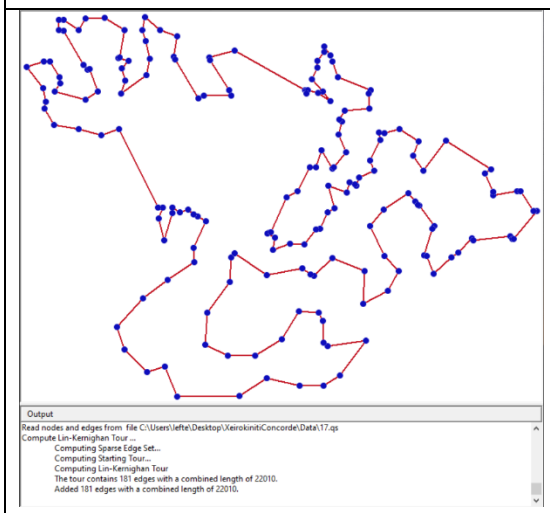
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

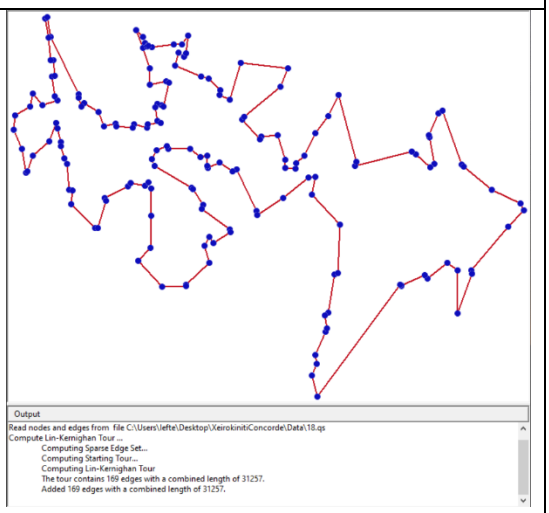
Σειρά: 16/17/18/19/20



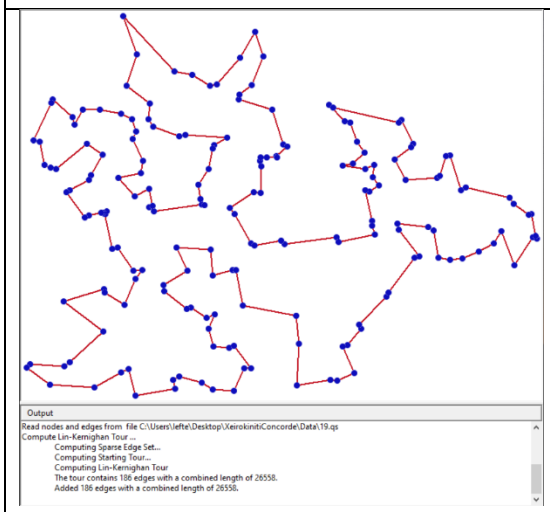
Συστάδα 16



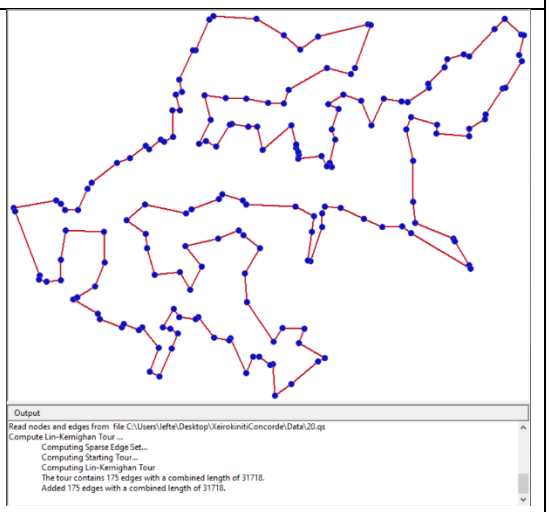
Συστάδα 17



Συστάδα 18



Συστάδα 19

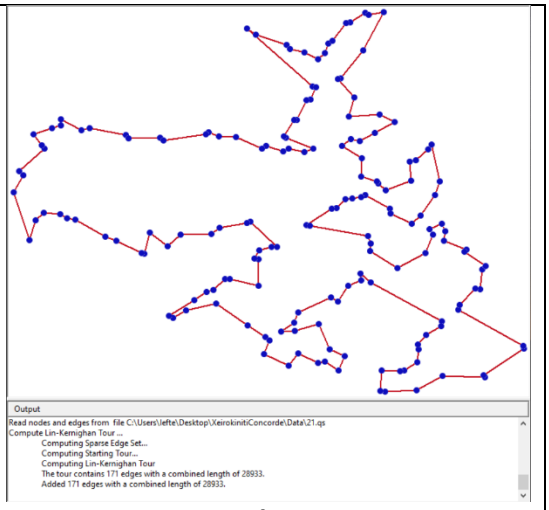


Συστάδα 20

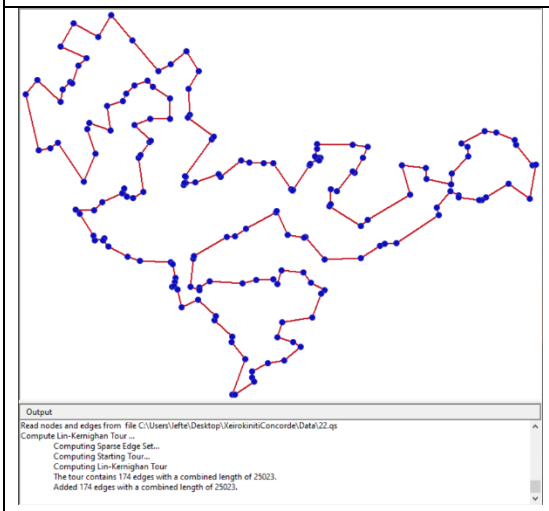
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

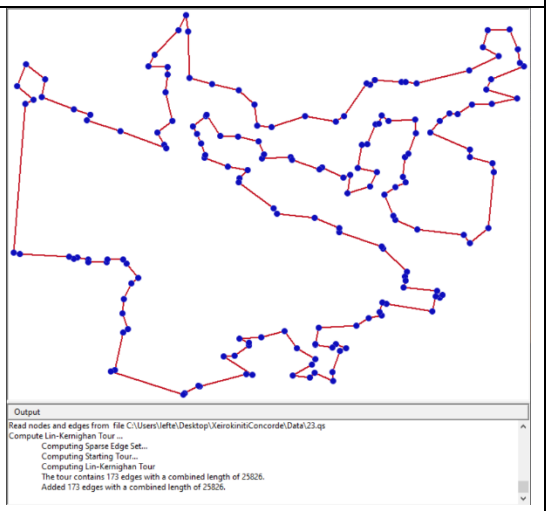
Σειρά: 21/22/23/24/25



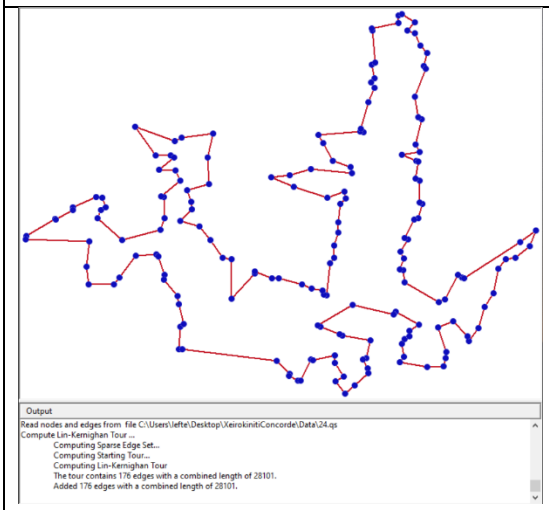
Συστάδα 21



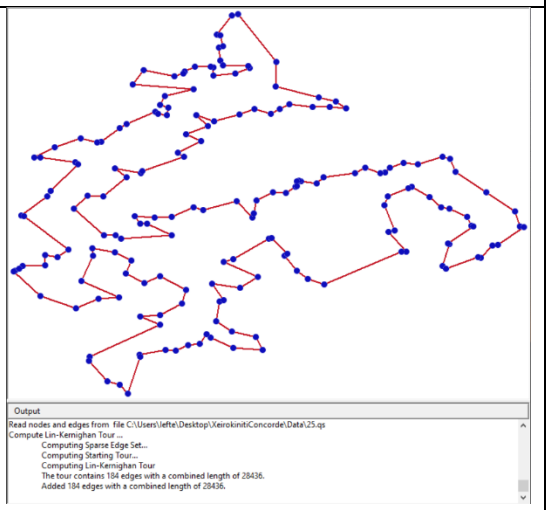
Συστάδα 22



Συστάδα 23



Συστάδα 24

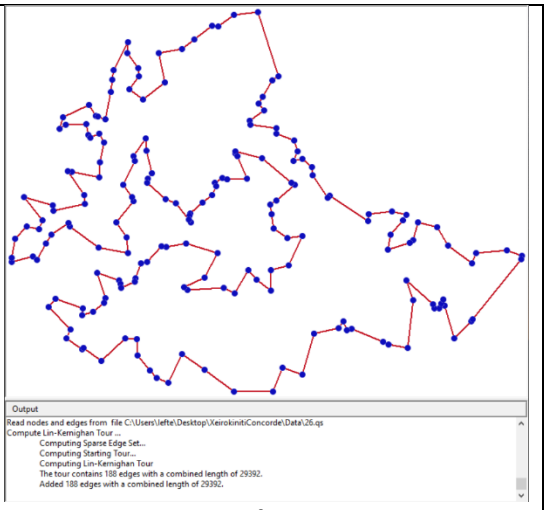


Συστάδα 25

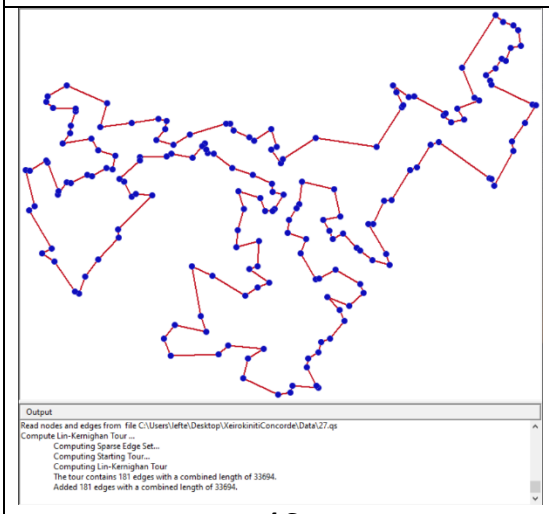
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

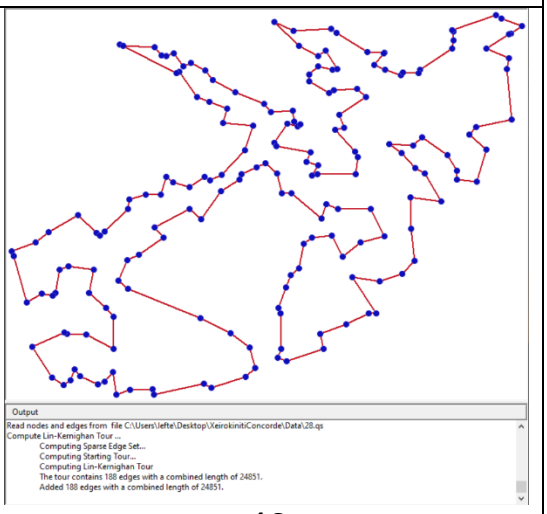
Σειρά: 26/27/28/29/30



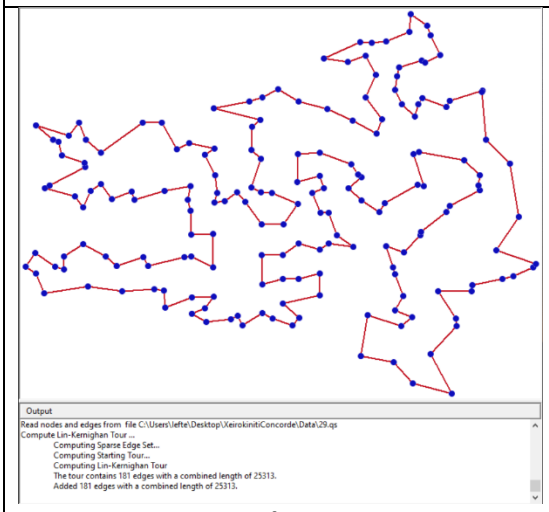
Συστάδα 26



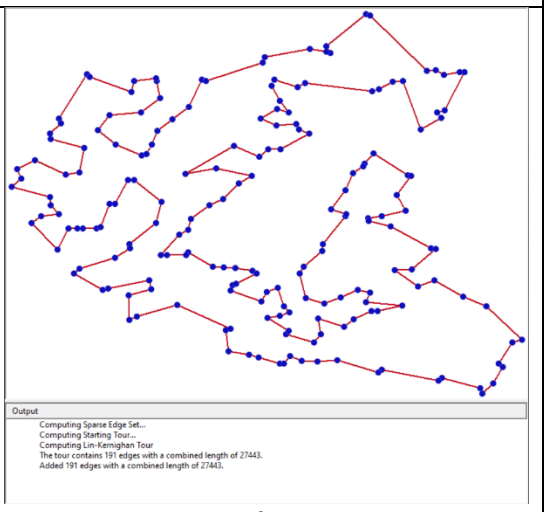
Συστάδα 27



Συστάδα 28



Συστάδα 29

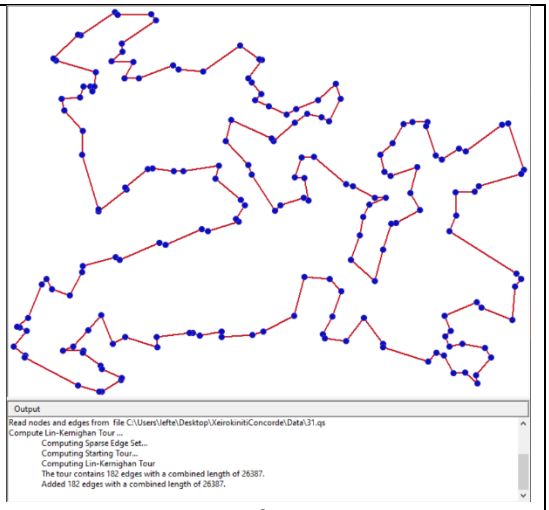


Συστάδα 30

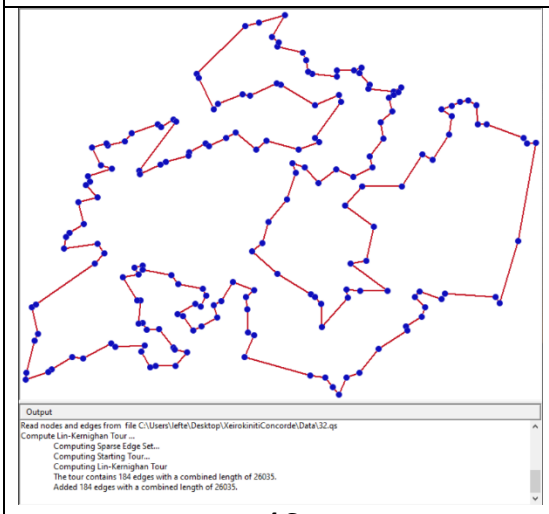
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

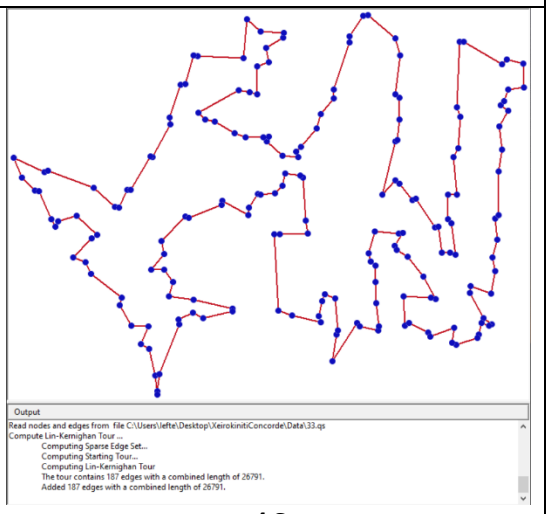
Σειρά: 31/32/33/34/35



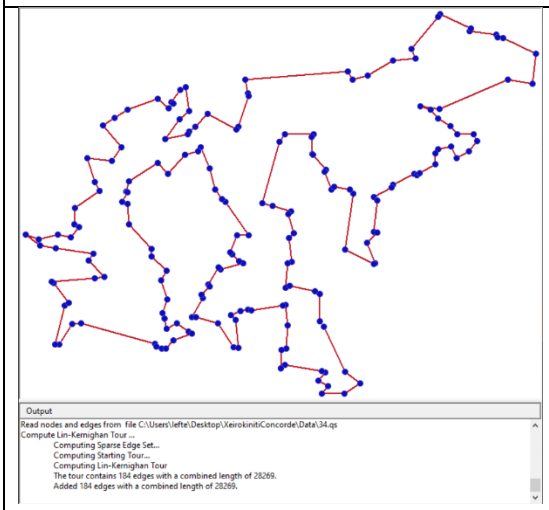
Συστάδα 31



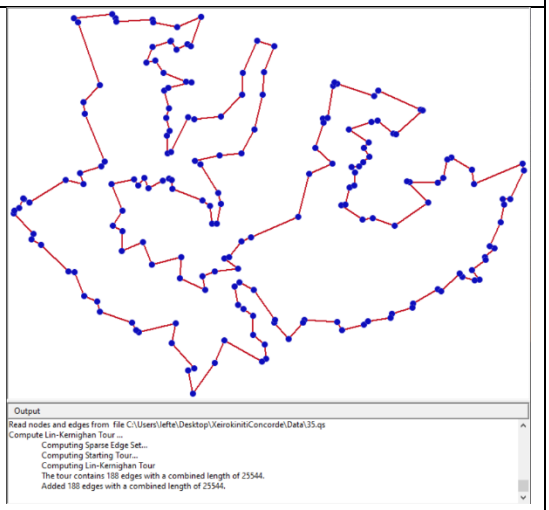
Συστάδα 32



Συστάδα 33



Συστάδα 34

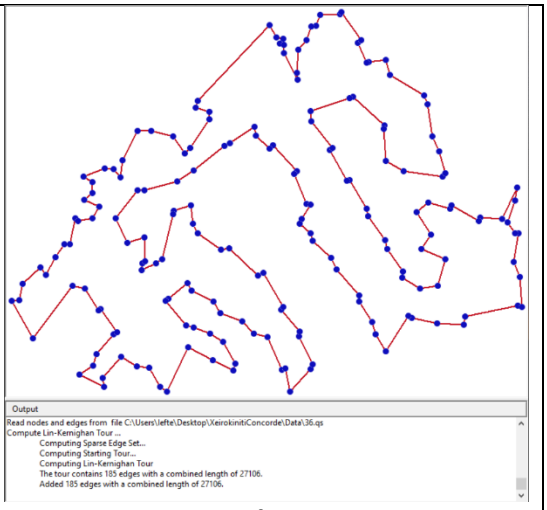


Συστάδα 35

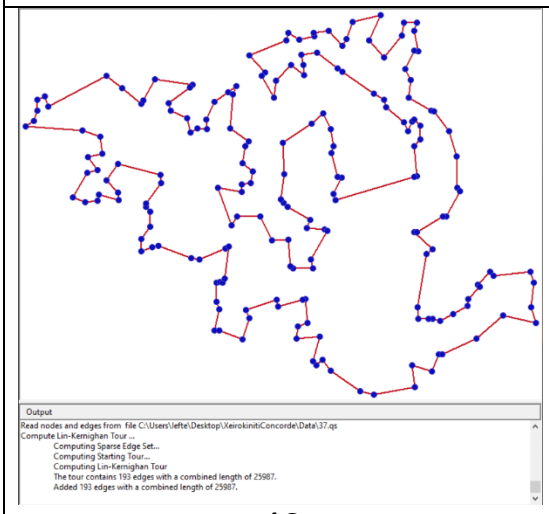
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

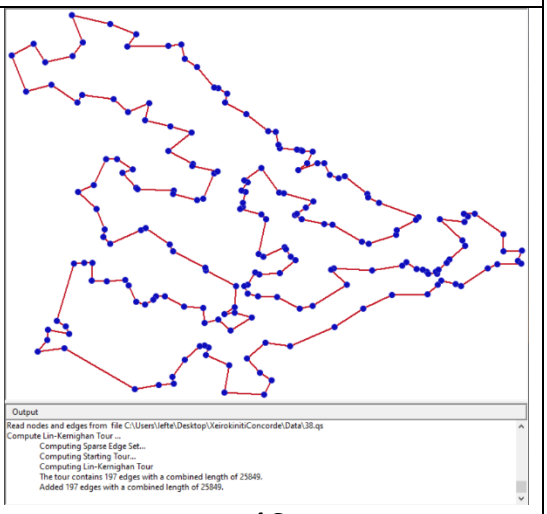
Σειρά: 36/37/38/39/40



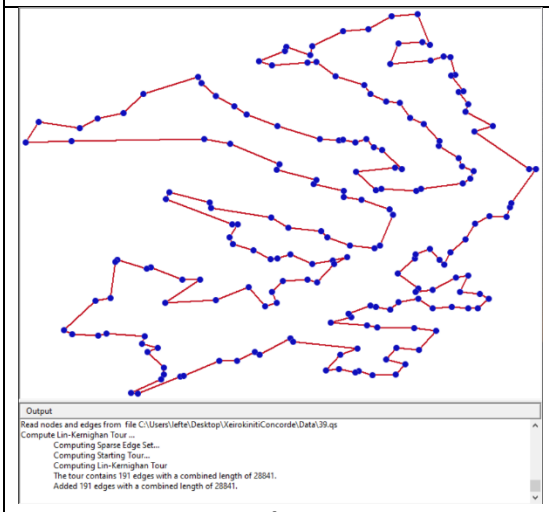
Συστάδα 36



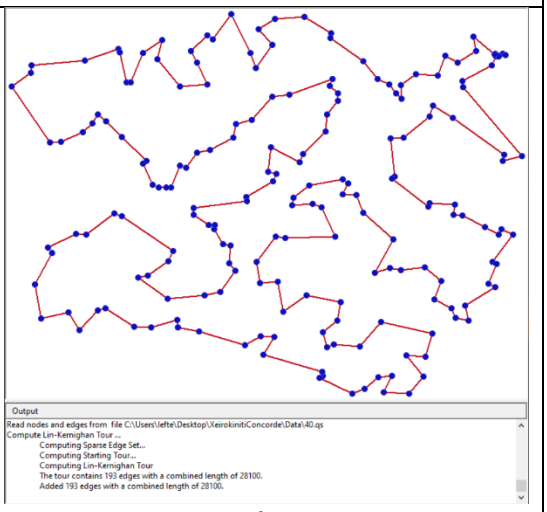
Συστάδα 37



Συστάδα 38



Συστάδα 39

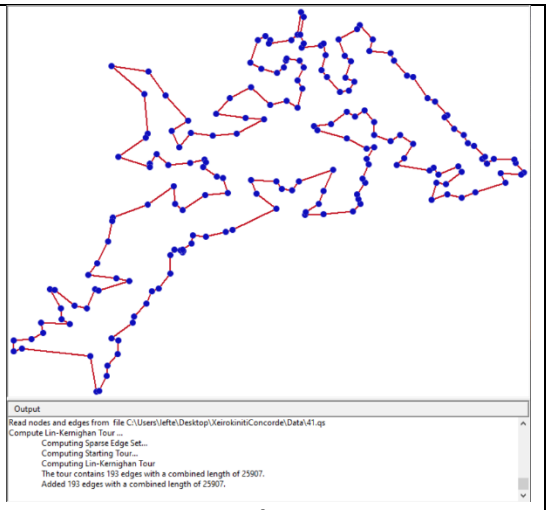


Συστάδα 40

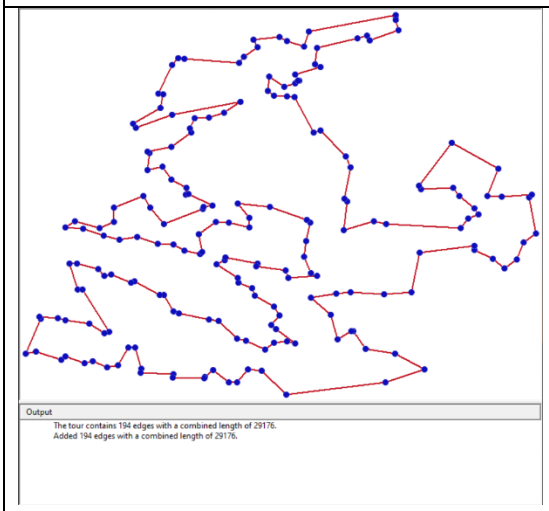
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

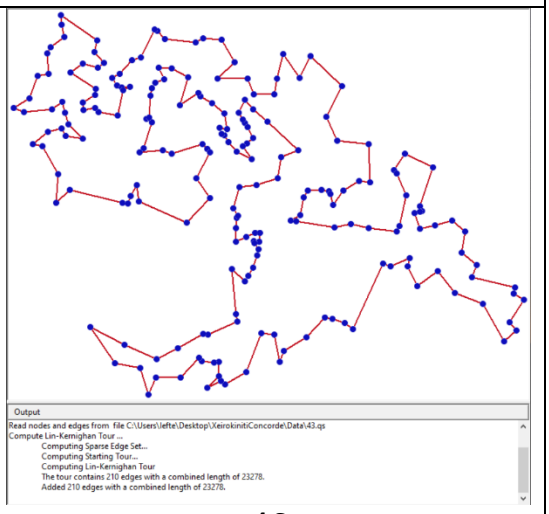
Σειρά: 41/42/43/44/45



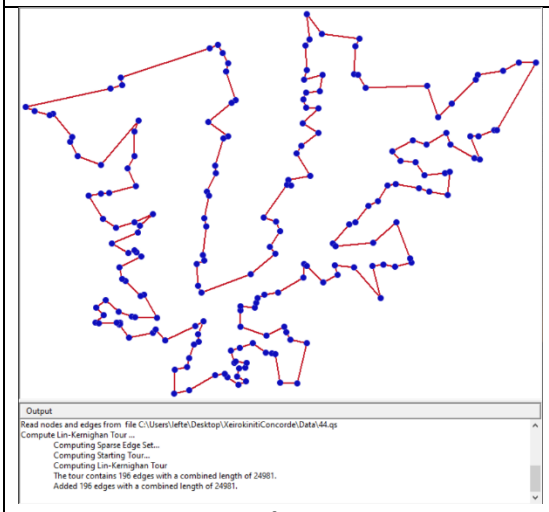
Συστάδα 41



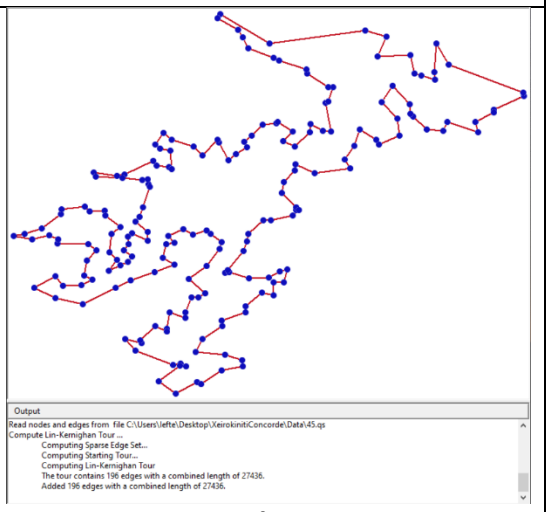
Συστάδα 42



Συστάδα 43



Συστάδα 44



Συστάδα 45

5.3.2 Σύνοψη αποτελεσμάτων υπερσυστάδων γρήγορης επίλυσης

Στον παρακάτω πίνακα παίρνουμε συγκεντρωτικά όλα τα αποτελέσματα που έδωσε ο Concorde TSP Solver και το χρόνο που απαιτεί η επιθεώρηση τους κατά την γρήγορη επίλυση:

Πίνακας 5.2 Αποτελέσματα υπερσυστάδων γρήγορης επίλυσης

Συστάδα	Στάσεις	Απόσταση	Χρόνος
1	125	50198	9.613033
2	142	40414	8.428767
3	151	46160	9.440667
4	145	42906	8.852567
5	158	39843	8.609783
6	154	35814	7.938767
7	172	35796	8.236067
8	169	43664	9.366267
9	162	38363	8.45445
10	162	32163	7.52445
11	165	24611	6.44165
12	172	26475	6.837917
13	168	31672	7.5508
14	172	27522	6.994967
15	189	29477	7.57155
16	165	33619	7.79285
17	181	22010	6.318167
18	169	31257	7.505217
19	186	26558	7.0837
20	175	31718	7.674367
21	171	28933	7.18995
22	174	25023	6.65345
23	173	25826	6.757233
24	176	28101	7.148483
25	184	28436	7.332067
26	188	29392	7.542133
27	181	33694	8.070767
28	188	24851	6.860983
29	181	25313	6.813617
30	191	27443	7.299783
31	182	26387	6.991383
32	184	26035	6.971917
33	187	26791	7.135317
34	184	28269	7.307017
35	188	25544	6.964933

36	185	27106	7.149233
37	193	25987	7.114717
38	197	25849	7.160683
39	191	28841	7.509483
40	193	28100	7.431667
41	193	25907	7.102717
42	194	29176	7.609733
43	210	23278	6.9917
44	196	24981	7.013817
45	196	27436	7.382067

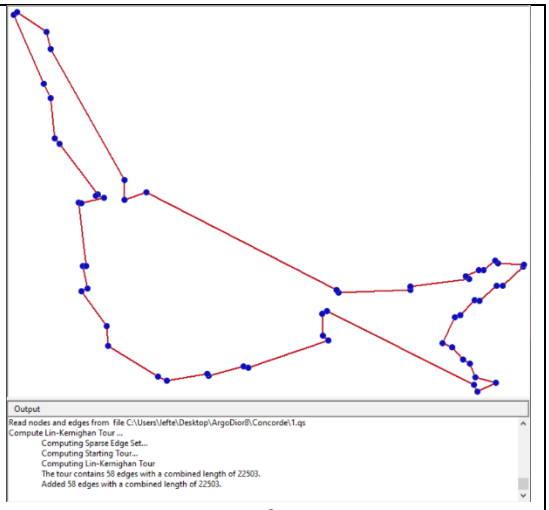
5.3.3 Αποτελέσματα Concorde TSP Solver για αργή επίλυση

Αφού εξάγουμε τις συντεταγμένες για κάθε μια από τις 89 υπερσυστάδες που δημιουργήσαμε για την αργή επίλυση και μορφώσουμε τα αρχεία .qs ώστε να εφαρμόσουμε τον αλγόριθμο Lin-Kernighan στο λογισμικό Concorde παίρνουμε τα εξής αποτελέσματα:

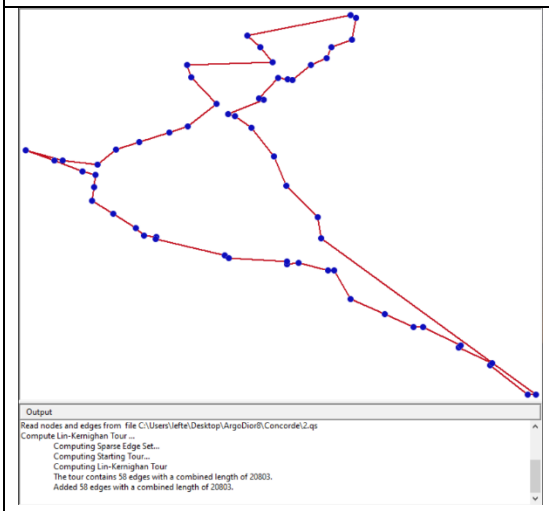
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

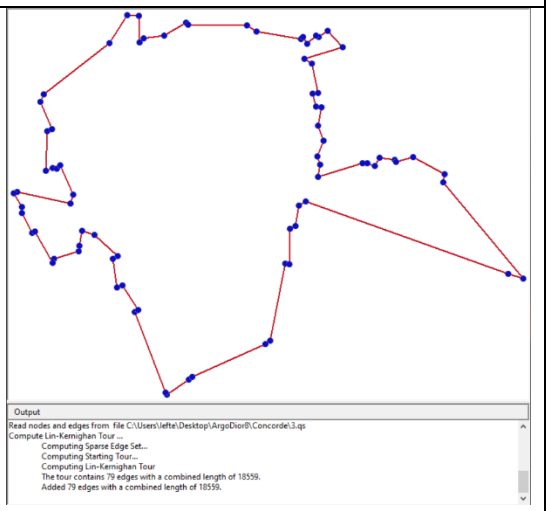
Σειρά: 1/2/3/4/5



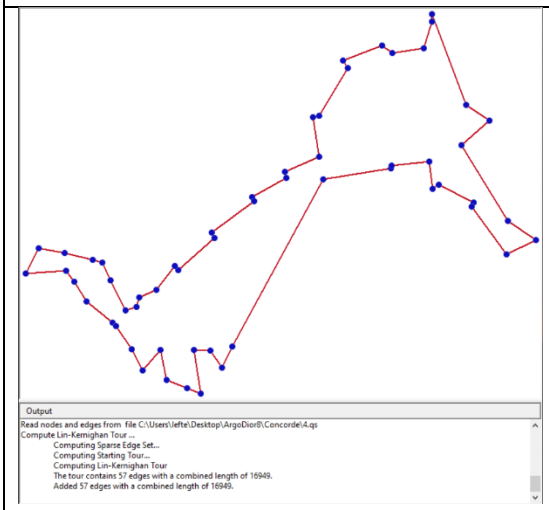
Συστάδα 1



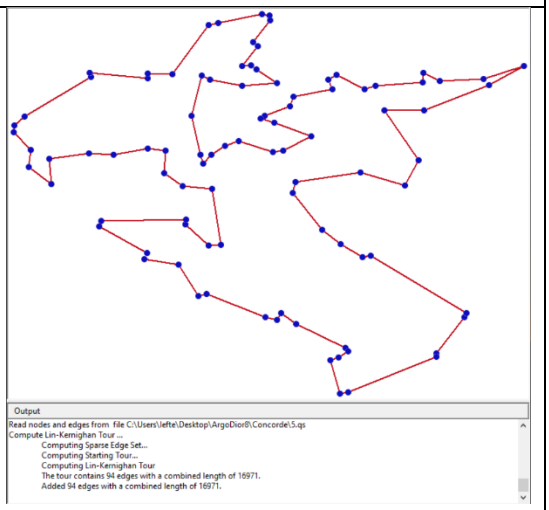
Συστάδα 2



Συστάδα 3



Συστάδα 4

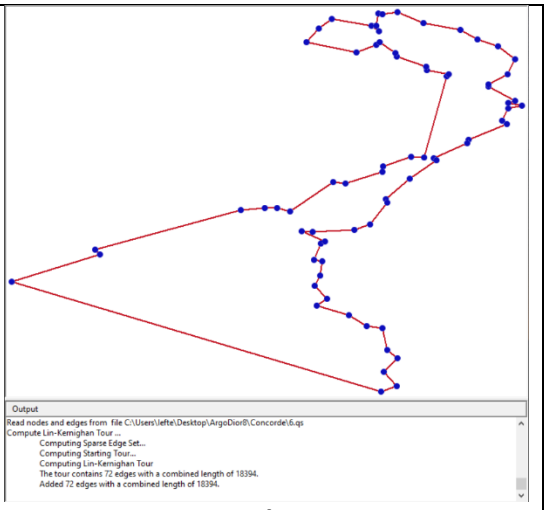


Συστάδα 5

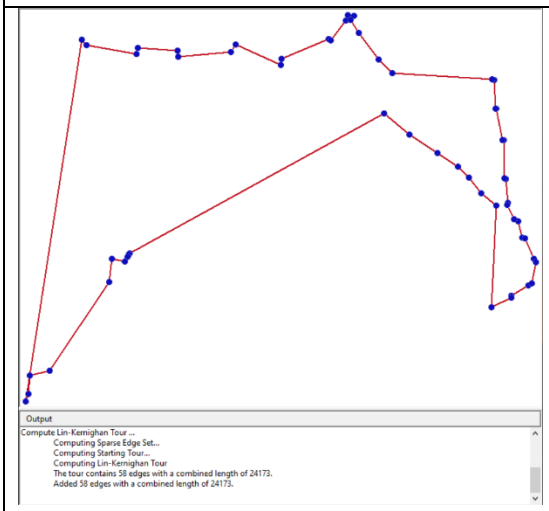
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

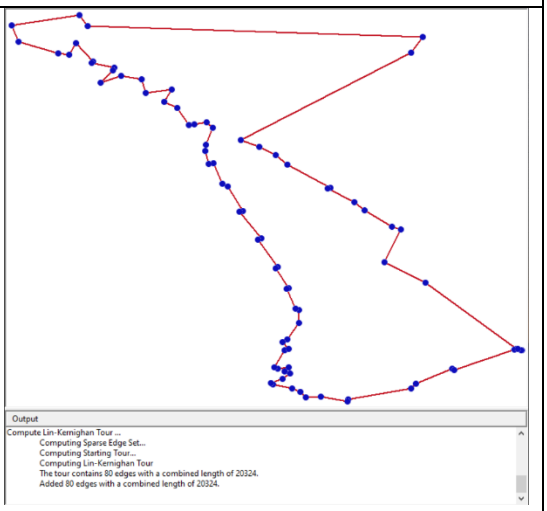
Σειρά: 6/7/8/9/10



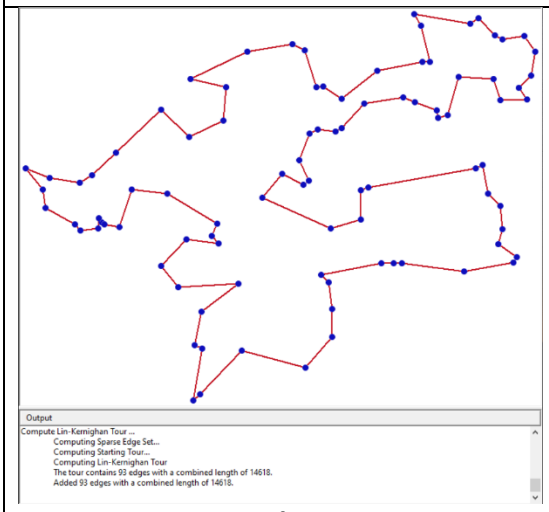
Συστάδα 6



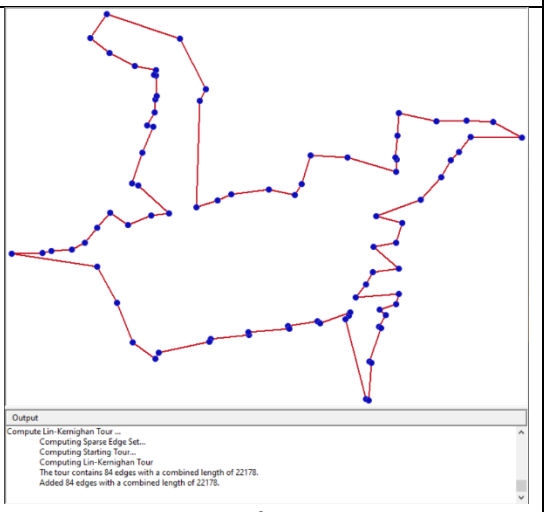
Συστάδα 7



Συστάδα 8



Συστάδα 9

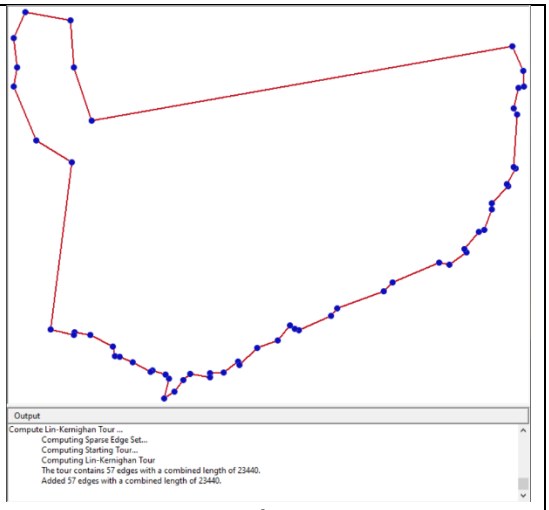


Συστάδα 10

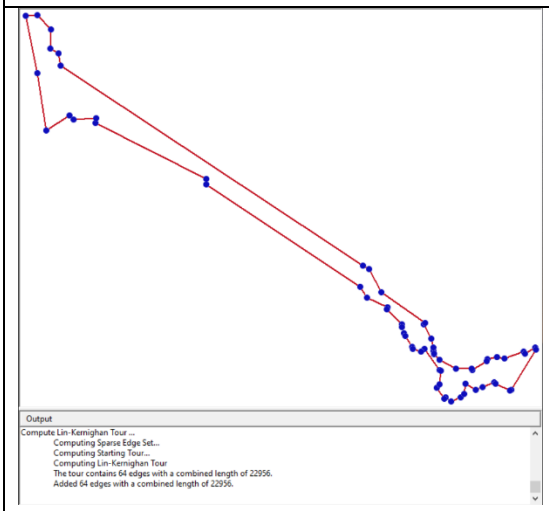
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

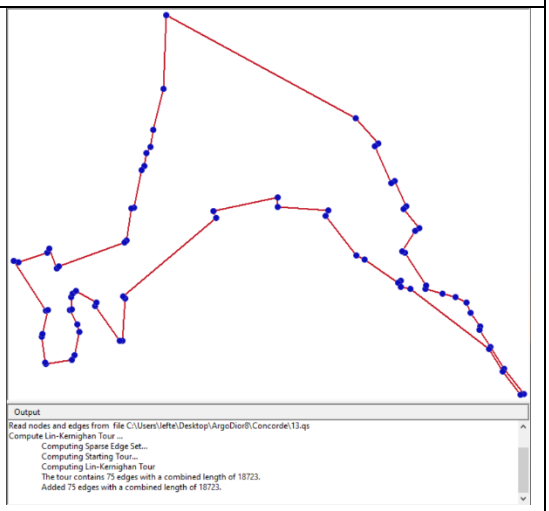
Σειρά: 11/12/13/14/15



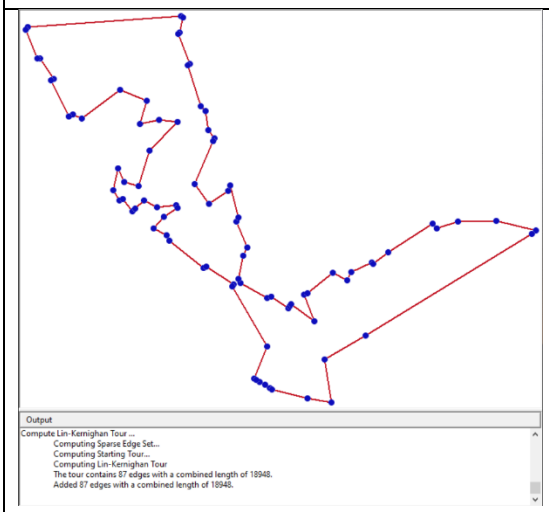
Συστάδα 11



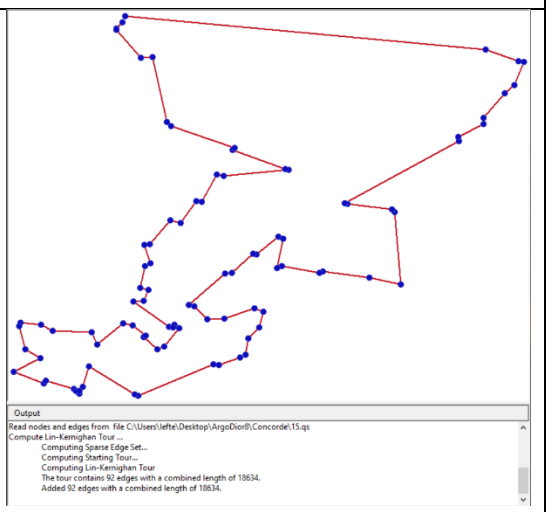
Συστάδα 12



Συστάδα 13



Συστάδα 14

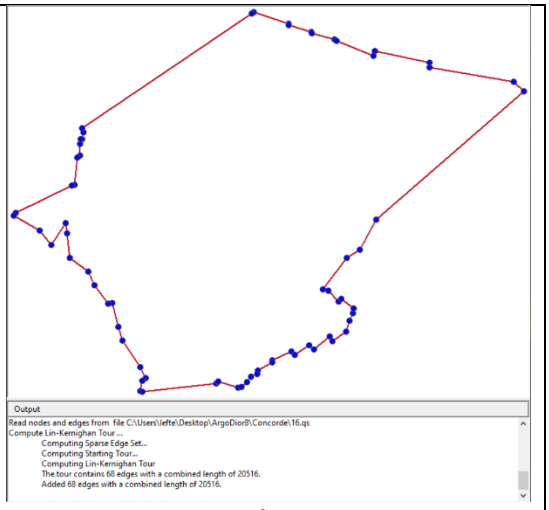


Συστάδα 15

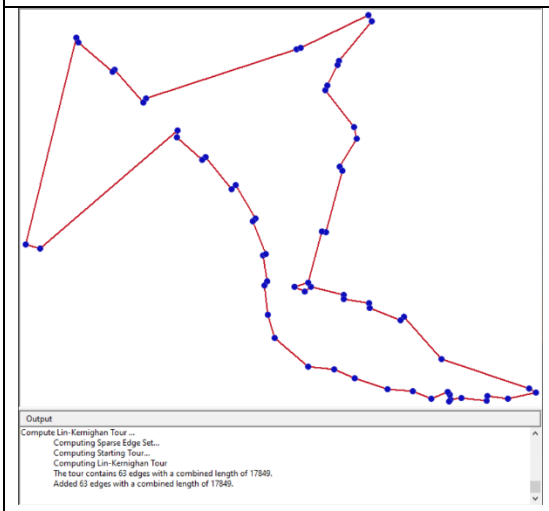
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

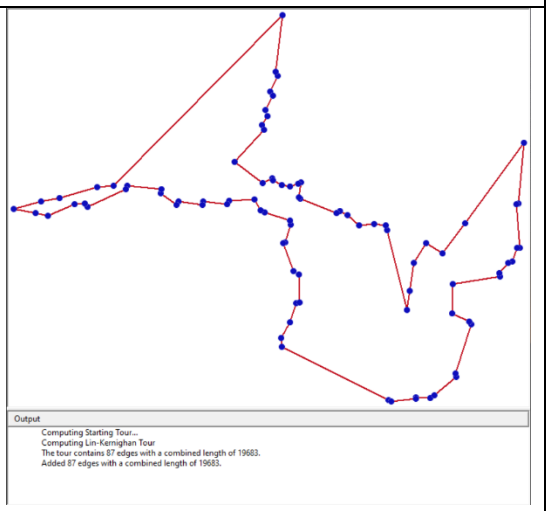
Σειρά: 16/17/18/19/20



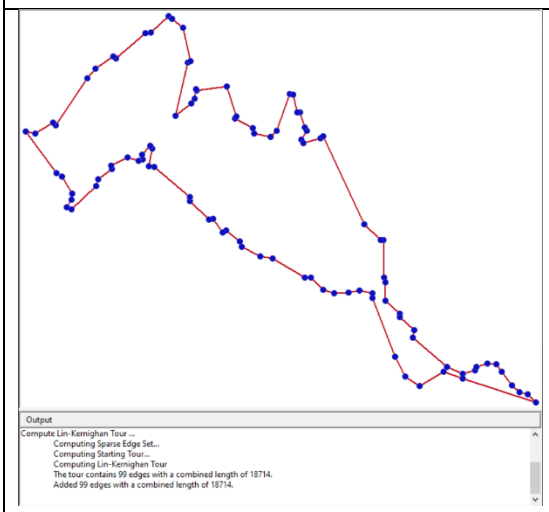
Συστάδα 16



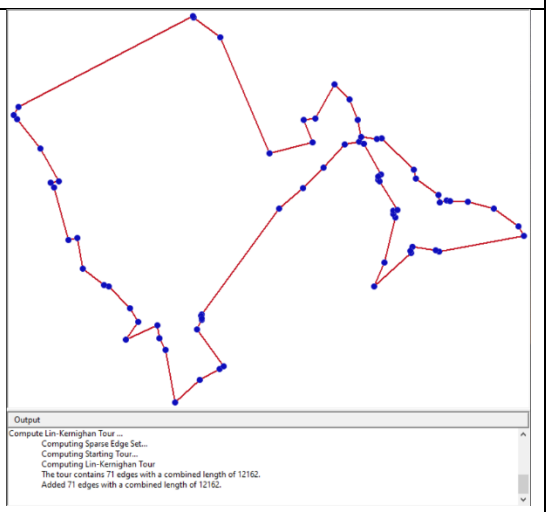
Συστάδα 17



Συστάδα 18



Συστάδα 19

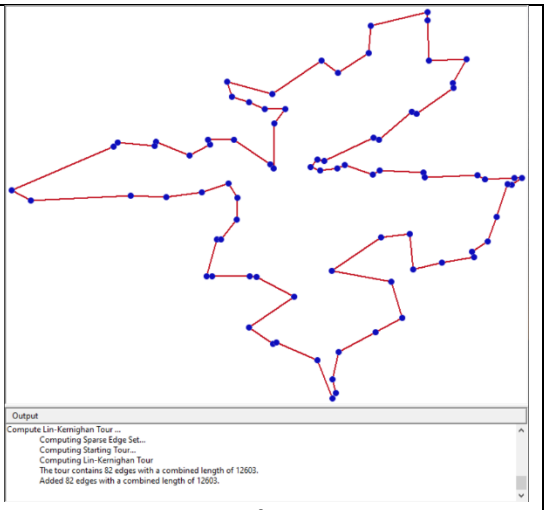


Συστάδα 20

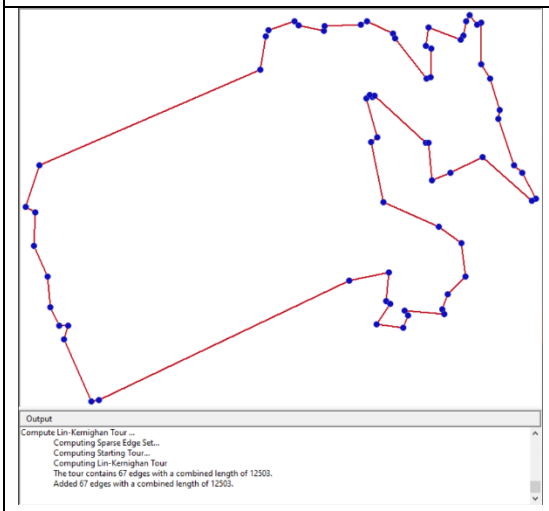
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

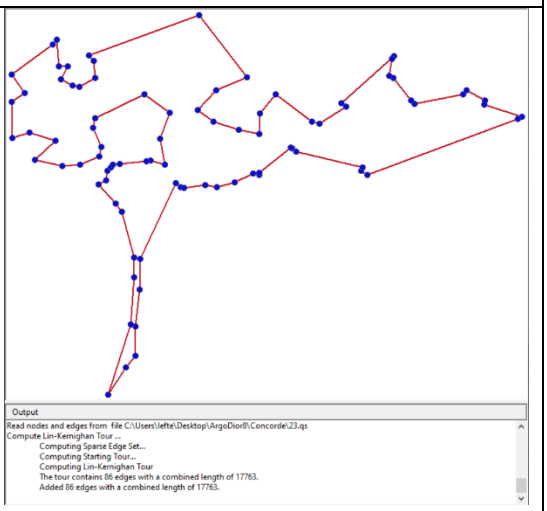
Σειρά: 21/22/23/24/25



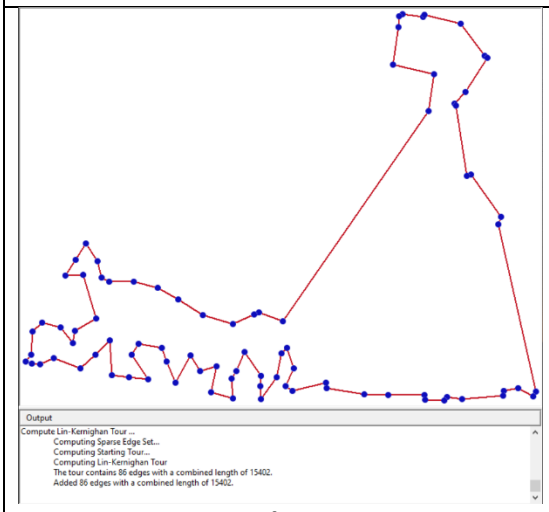
Συστάδα 21



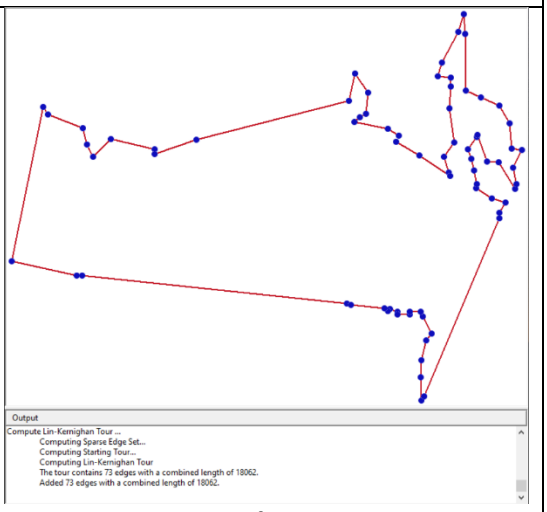
Συστάδα 22



Συστάδα 23



Συστάδα 24

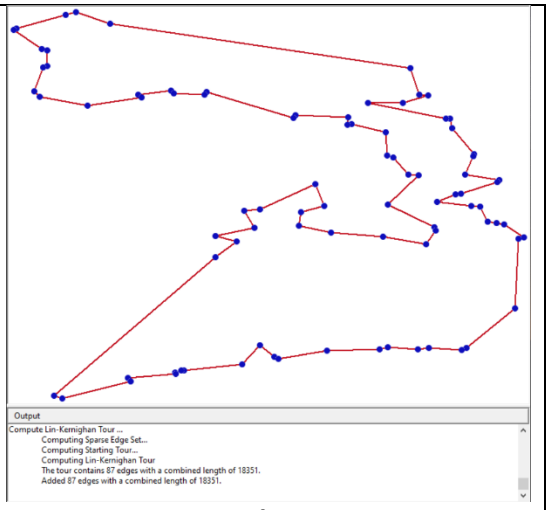


Συστάδα 25

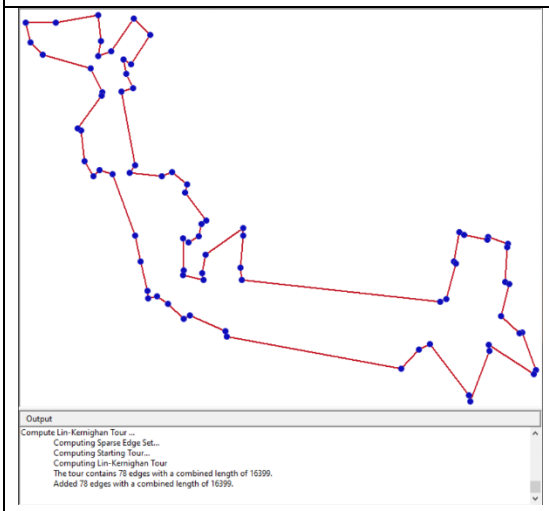
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

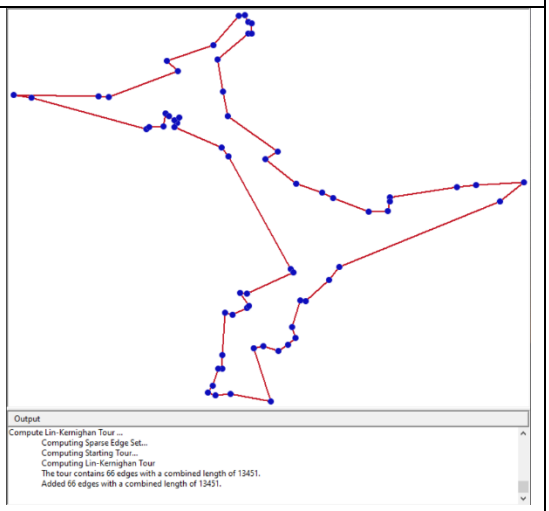
Σειρά: 26/27/28/29/30



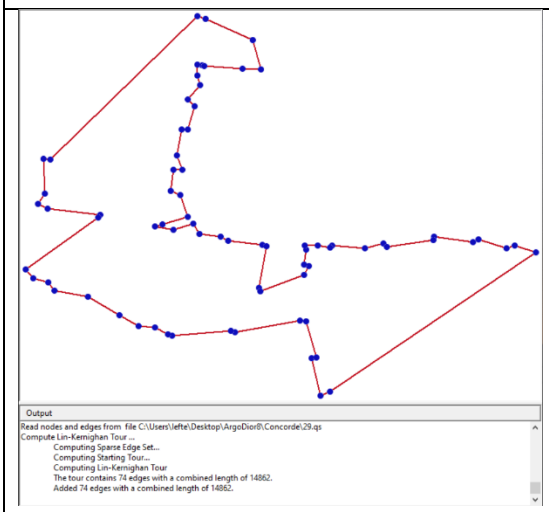
Συστάδα 26



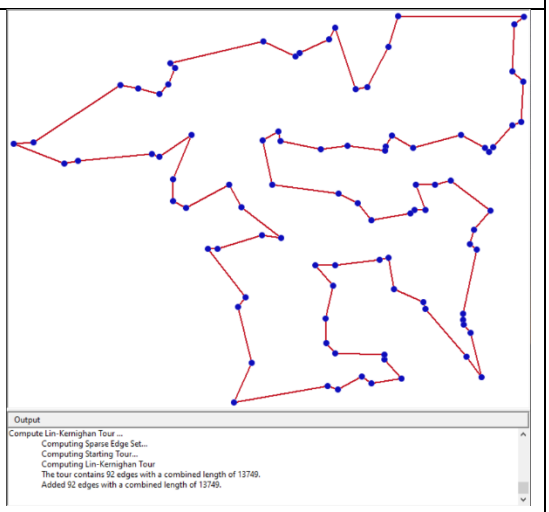
Συστάδα 27



Συστάδα 28



Συστάδα 29

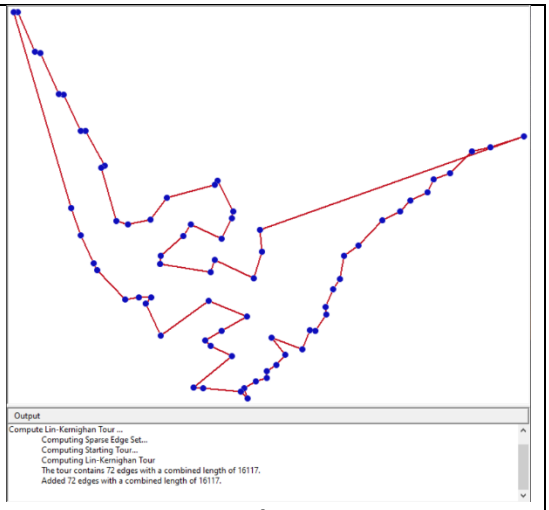


Συστάδα 30

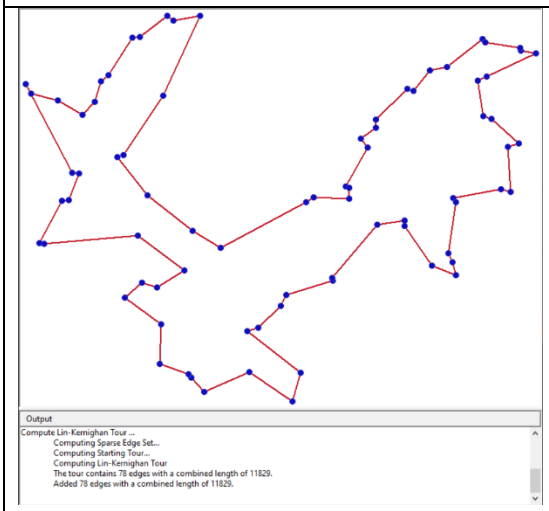
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

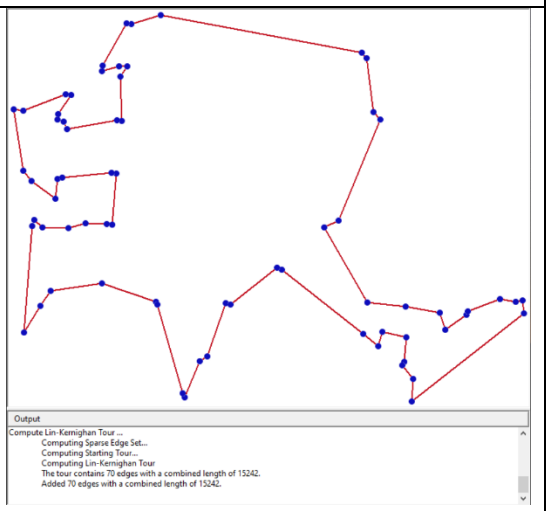
Σειρά: 31/32/33/34/35



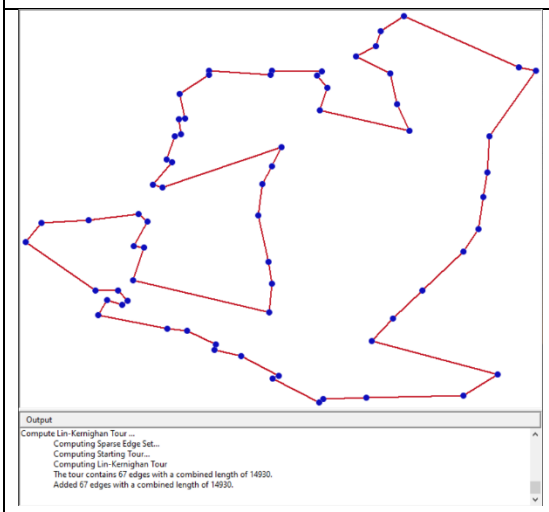
Συστάδα 31



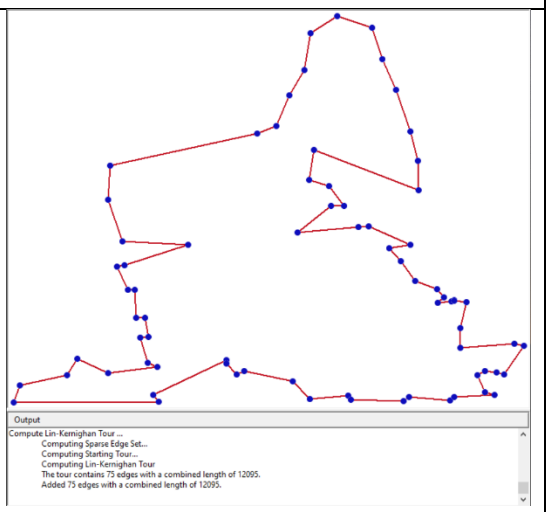
Συστάδα 32



Συστάδα 33



Συστάδα 34

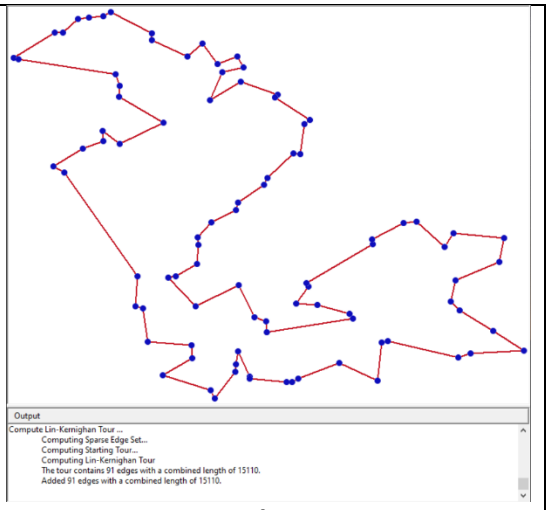


Συστάδα 35

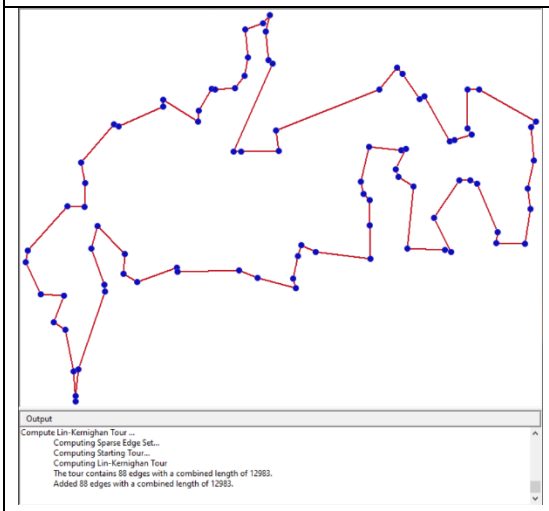
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

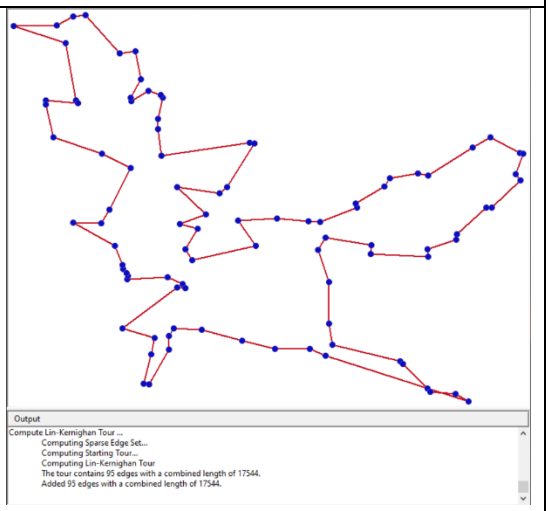
Σειρά: 36/37/38/39/40



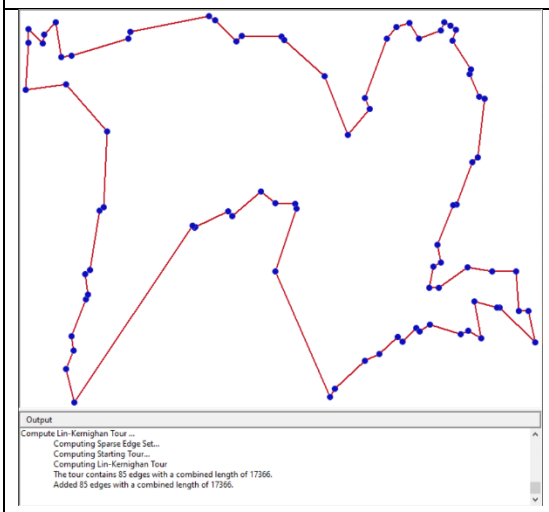
Συστάδα 36



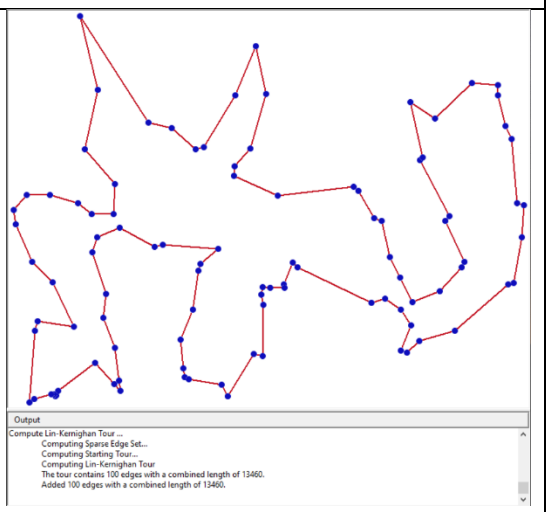
Συστάδα 37



Συστάδα 38



Συστάδα 39

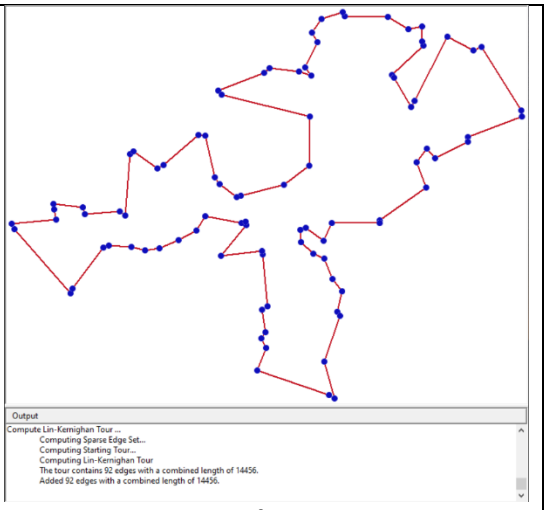


Συστάδα 40

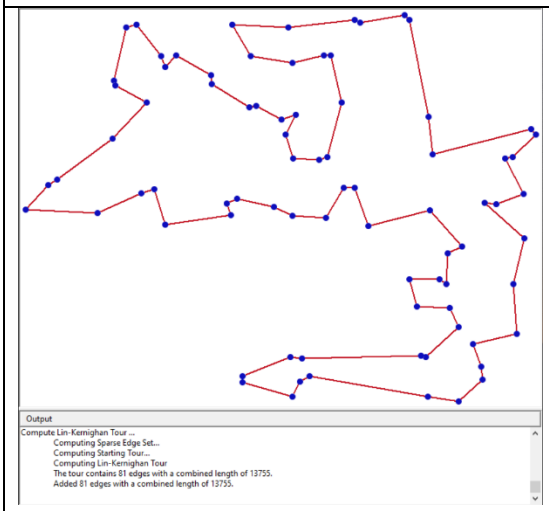
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

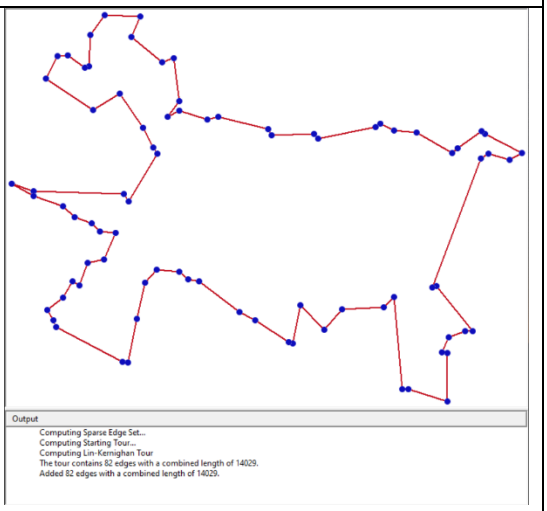
Σειρά: 41/42/43/44/45



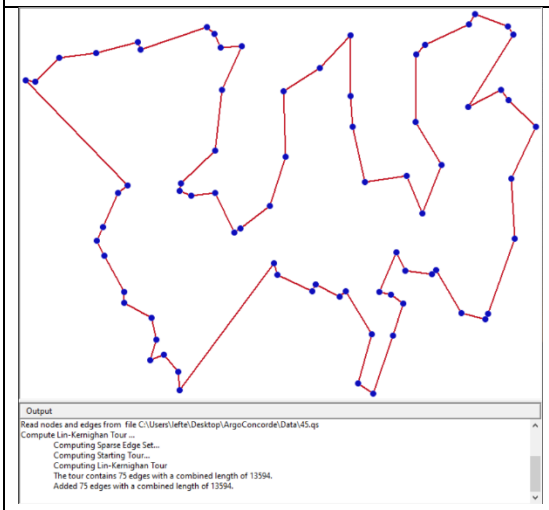
Συστάδα 41



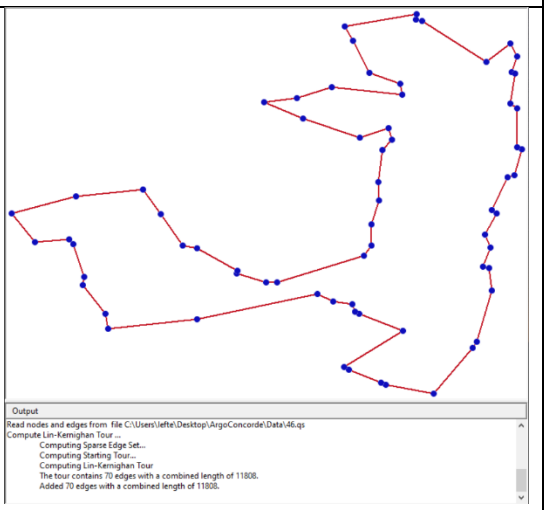
Συστάδα 42



Συστάδα 43



Συστάδα 44

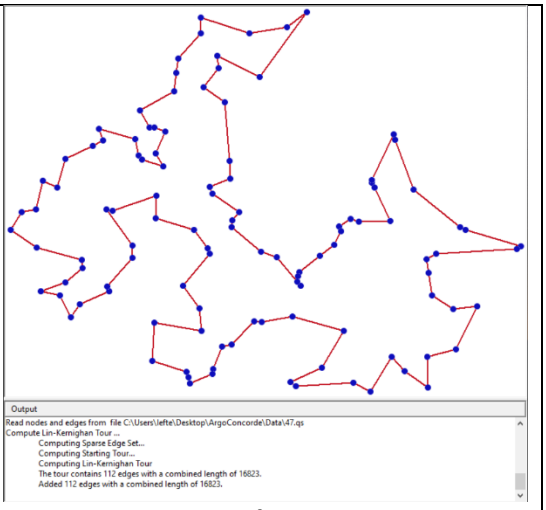


Συστάδα 45

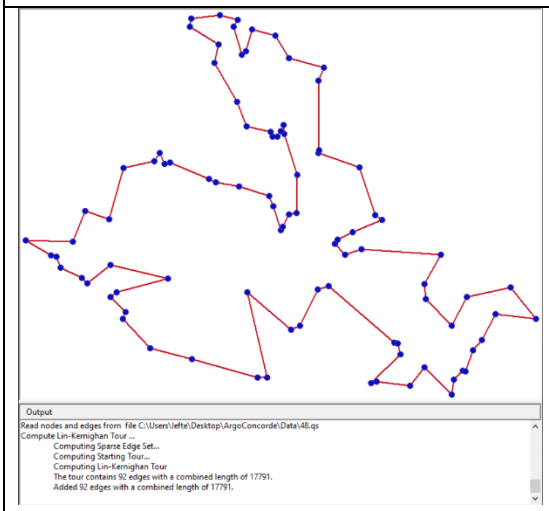
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

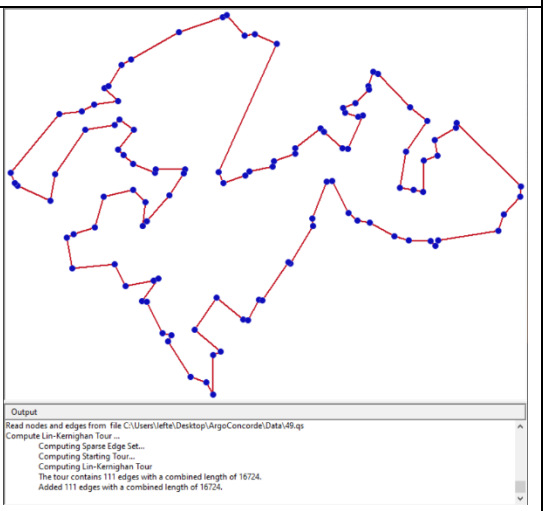
Σειρά: 46/47/48/49/50



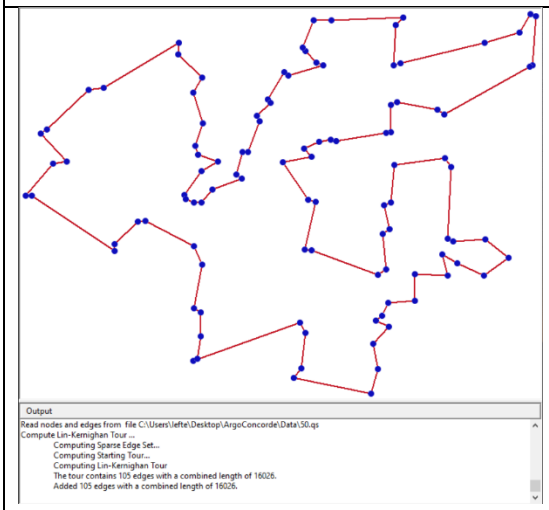
Συστάδα 46



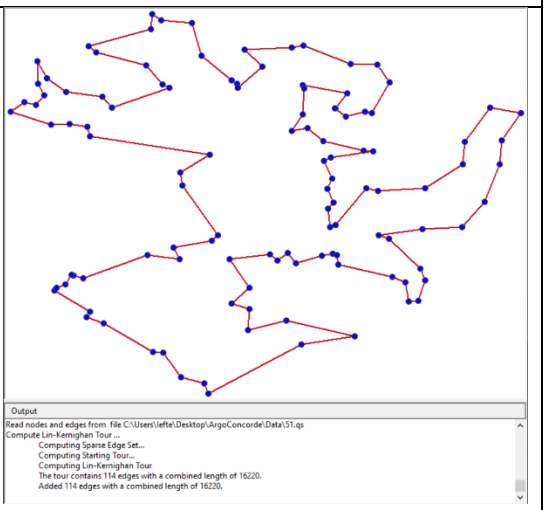
Συστάδα 47



Συστάδα 48



Συστάδα 49

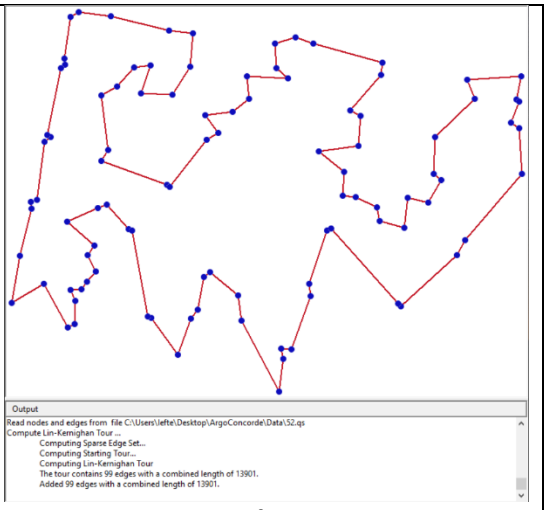


Συστάδα 50

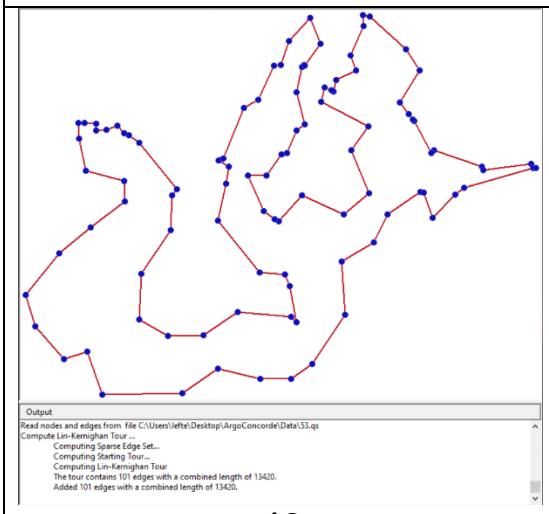
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

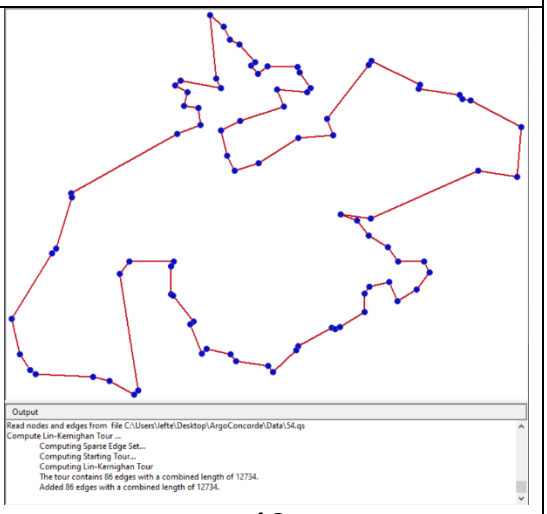
Σειρά: 51/52/53/54/55



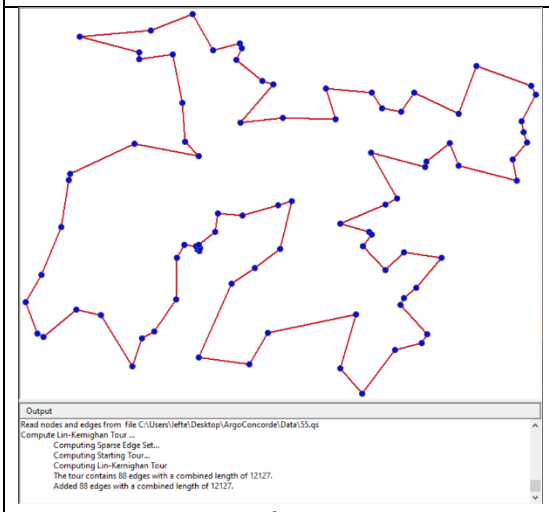
Συστάδα 51



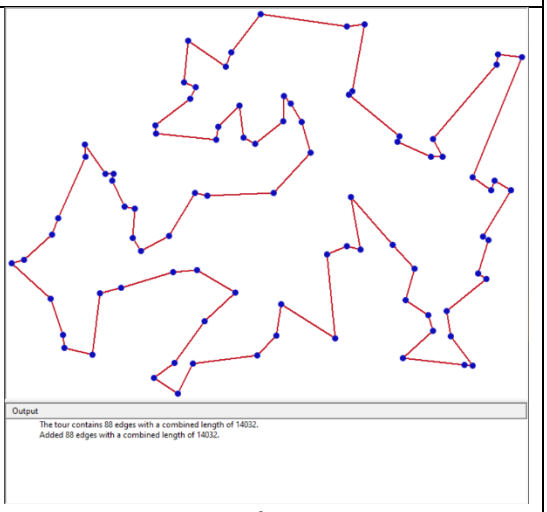
Συστάδα 52



Συστάδα 53



Συστάδα 54

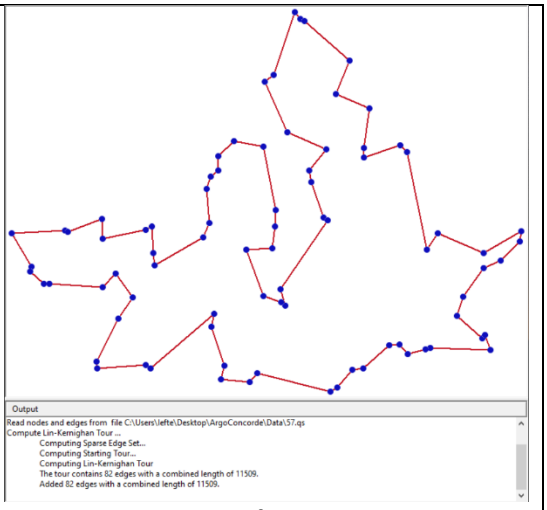


Συστάδα 55

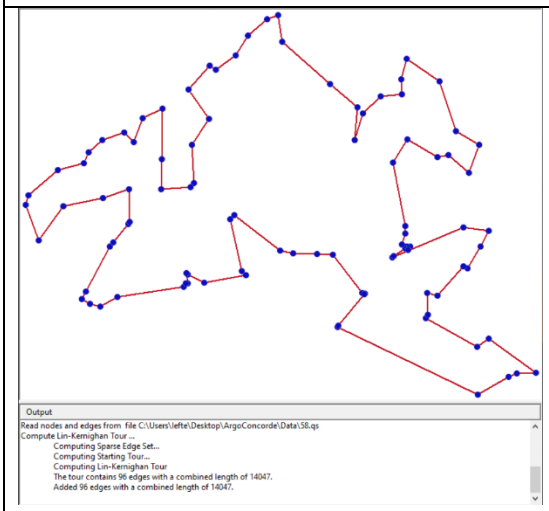
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

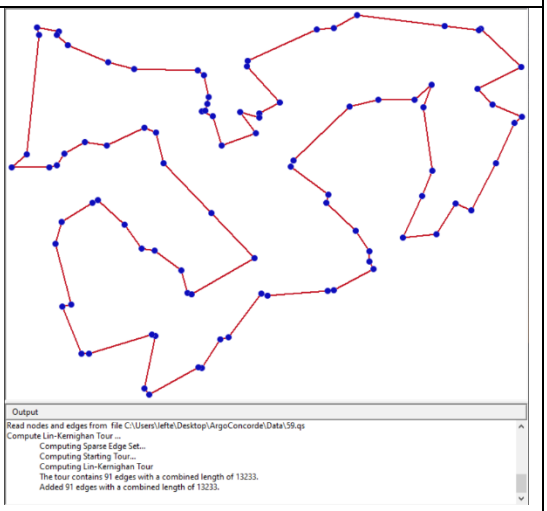
Σειρά: 56/57/58/59/60



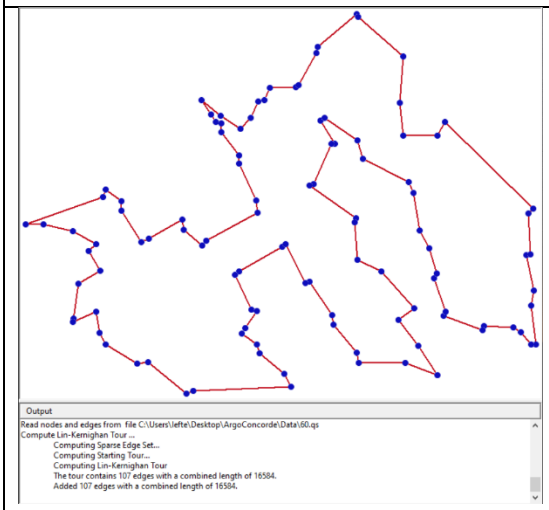
Συστάδα 56



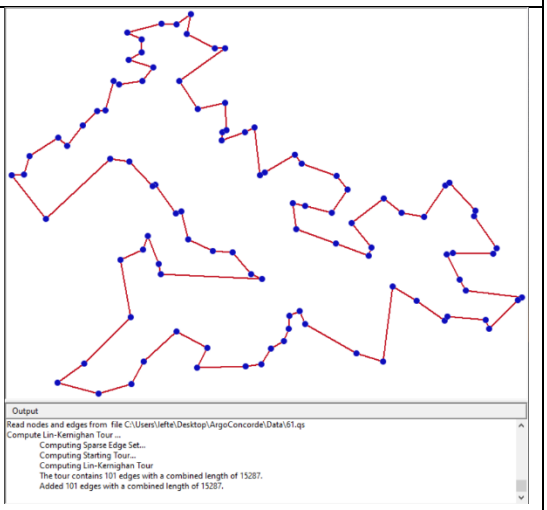
Συστάδα 57



Συστάδα 58



Συστάδα 59

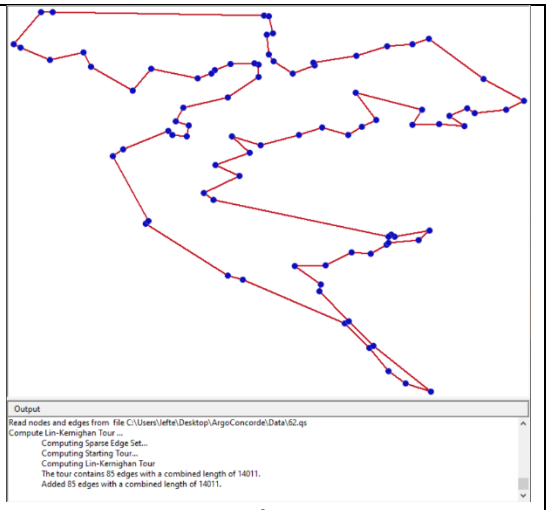


Συστάδα 60

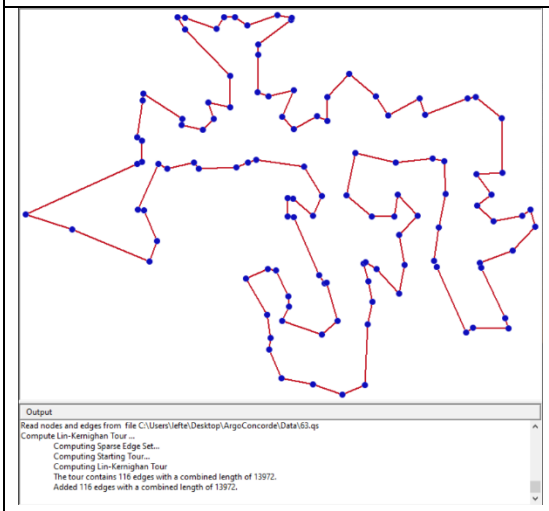
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

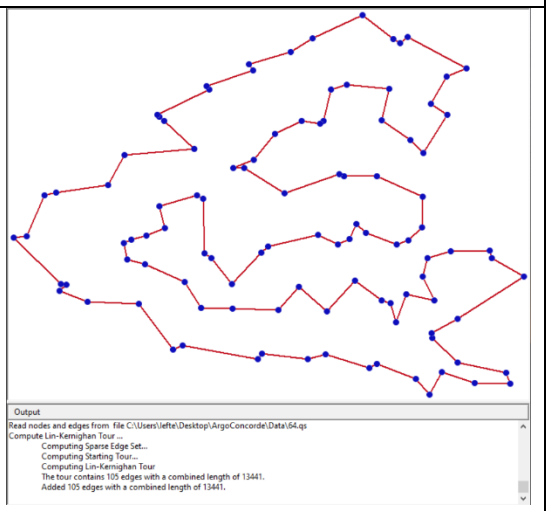
Σειρά: 61/62/63/64/65



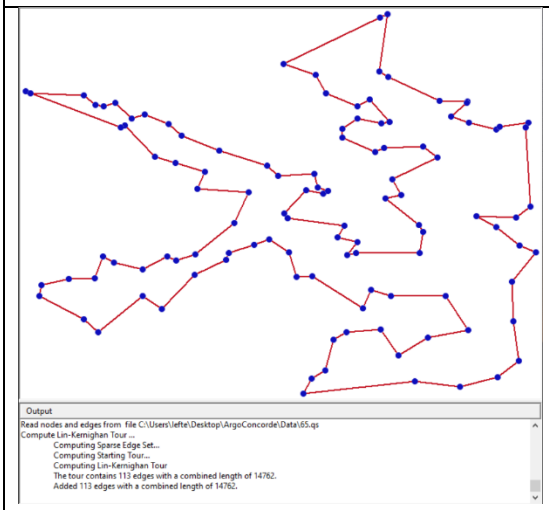
Συστάδα 61



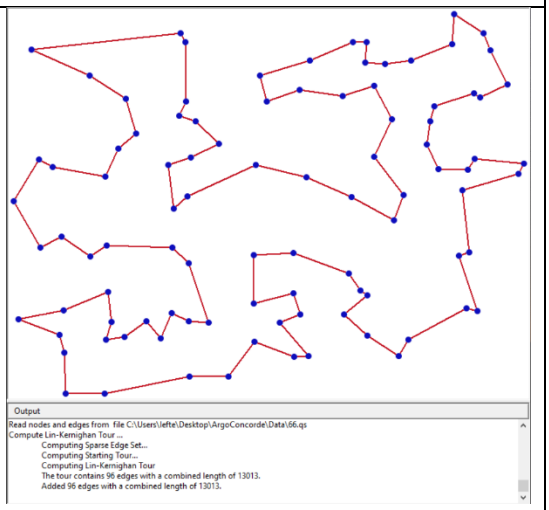
Συστάδα 62



Συστάδα 63



Συστάδα 64

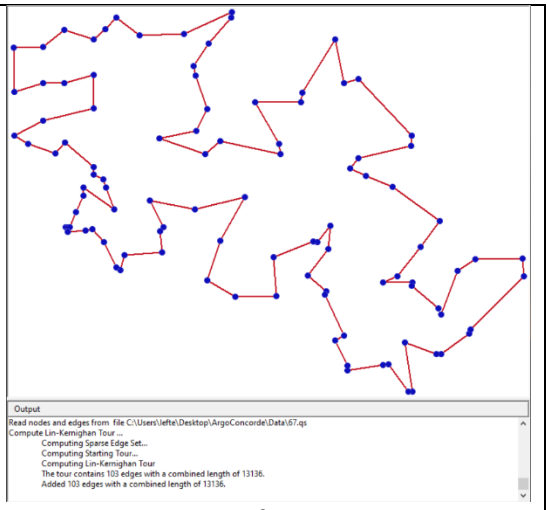


Συστάδα 65

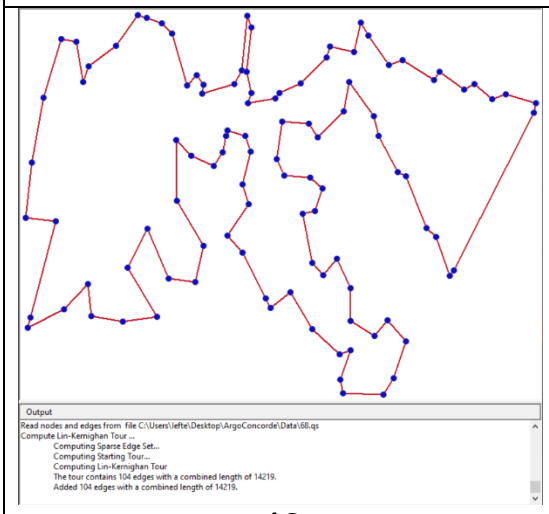
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

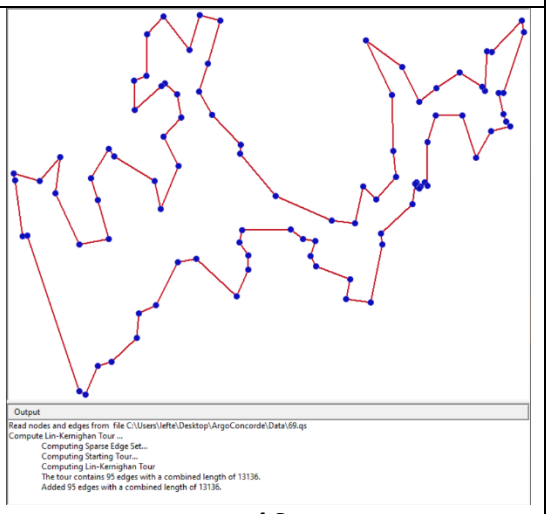
Σειρά: 66/67/68/69/70



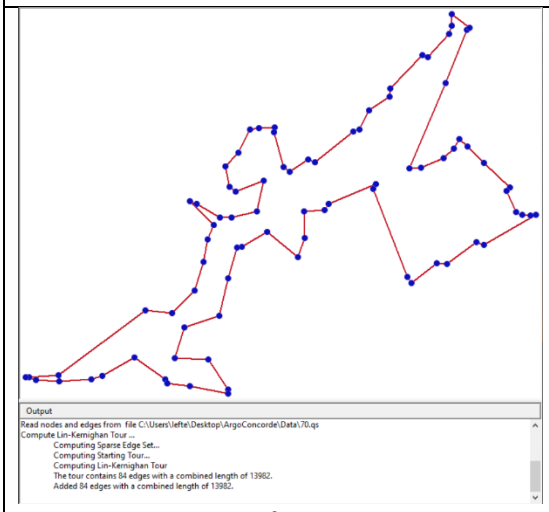
Συστάδα 66



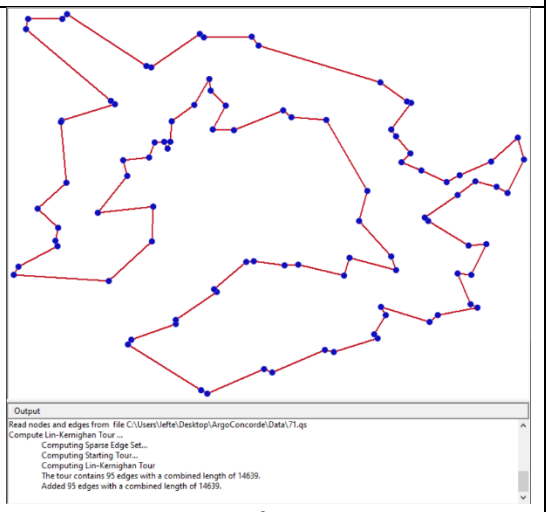
Συστάδα 67



Συστάδα 68



Συστάδα 69

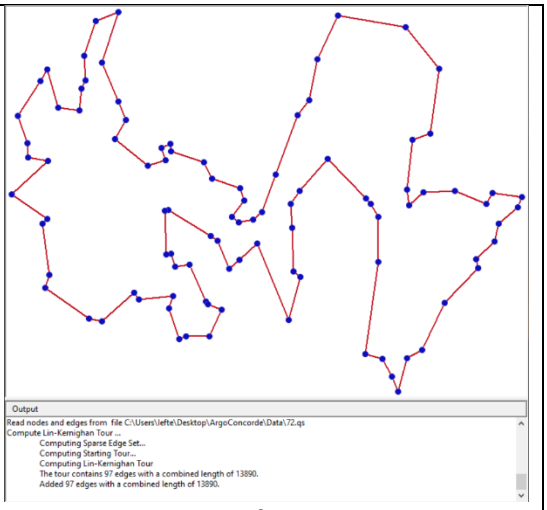


Συστάδα 70

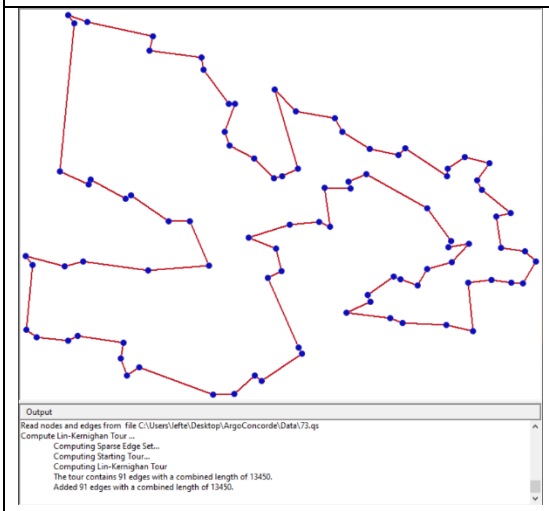
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

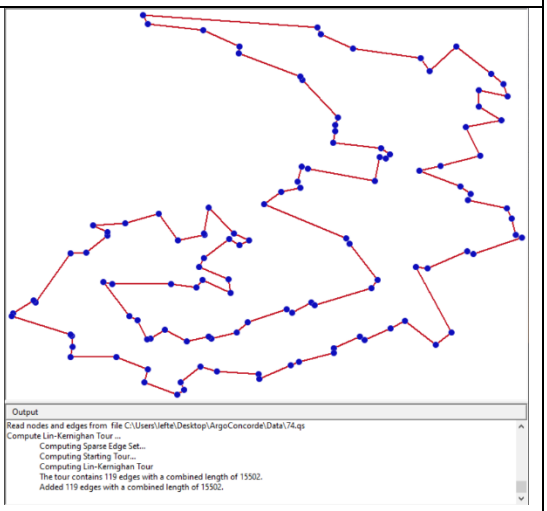
Σειρά: 71/72/73/74/75



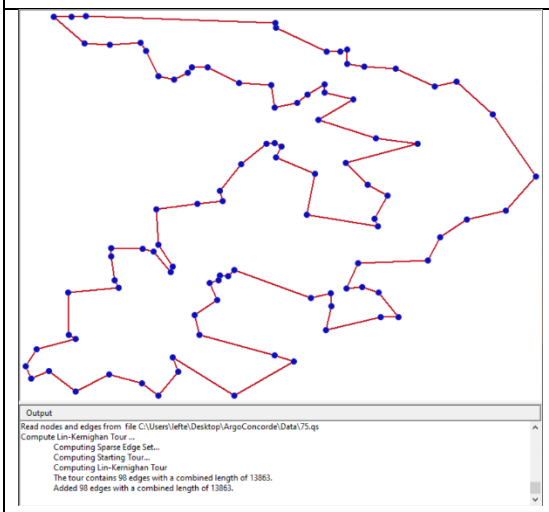
Συστάδα 71



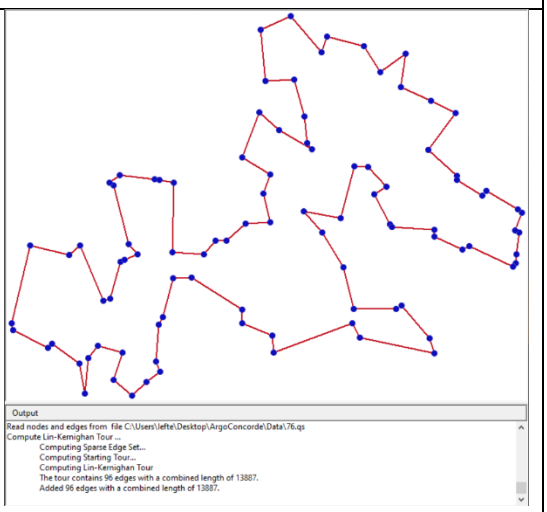
Συστάδα 72



Συστάδα 73



Συστάδα 74

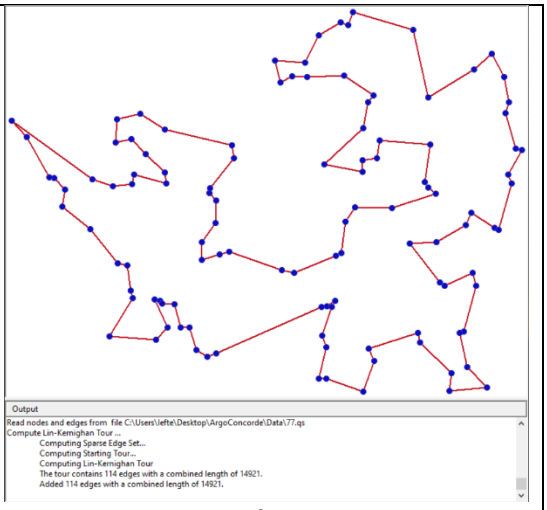


Συστάδα 75

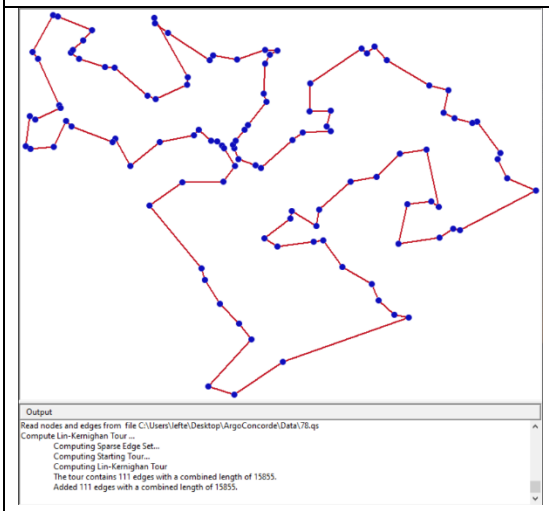
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

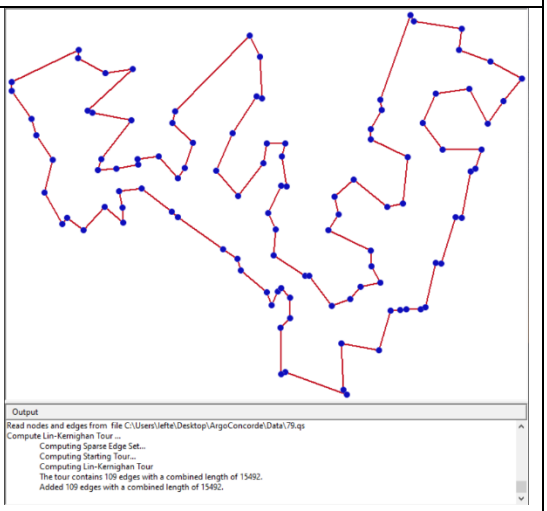
Σειρά: 76/77/78/79/80



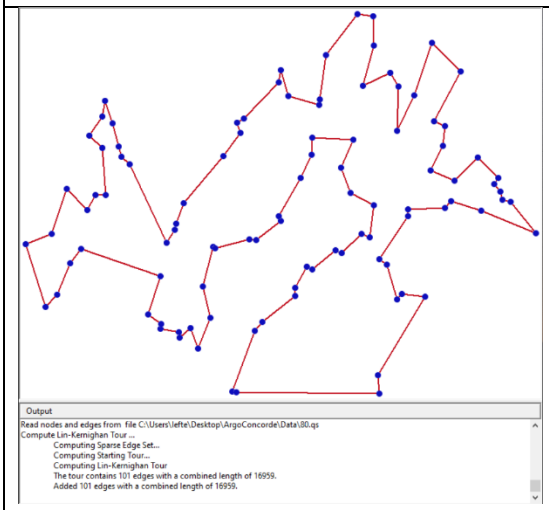
Συστάδα 76



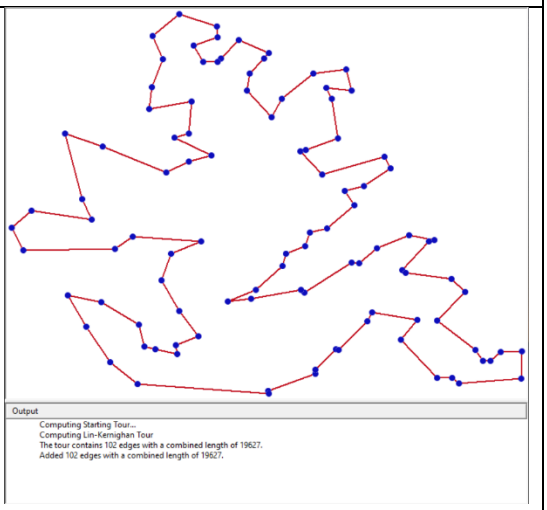
Συστάδα 77



Συστάδα 78



Συστάδα 79

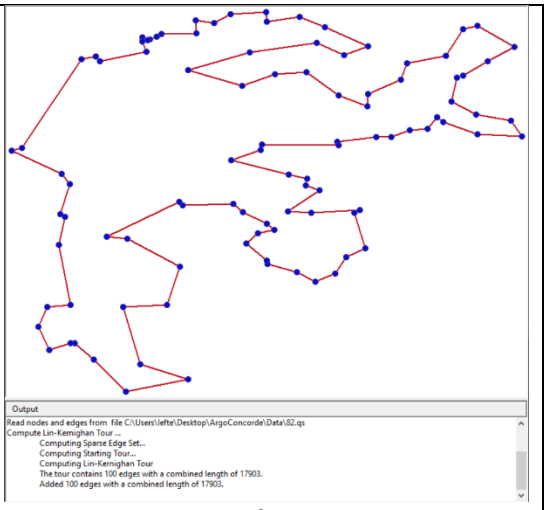


Συστάδα 80

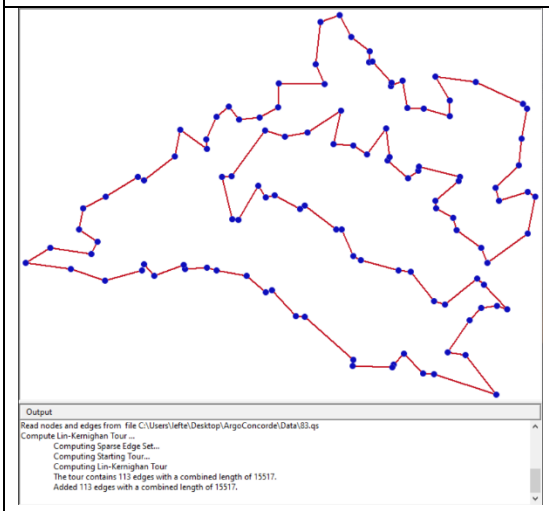
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

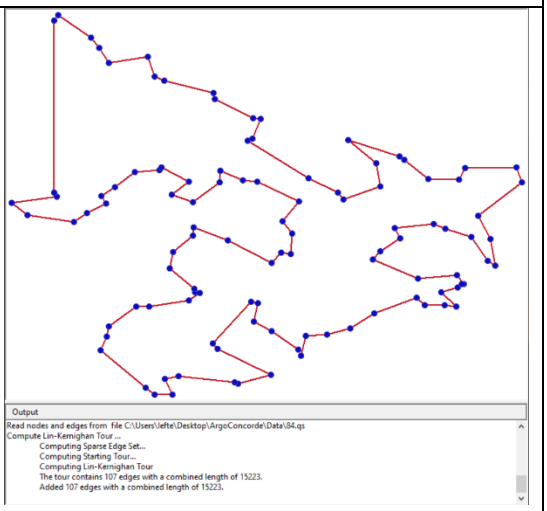
Σειρά: 81/82/83/84/85



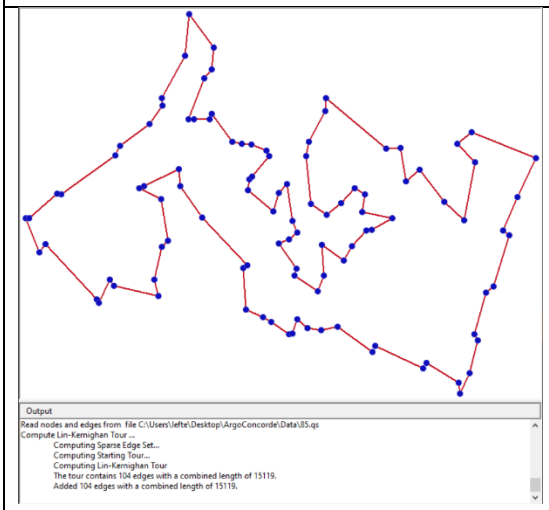
Συστάδα 81



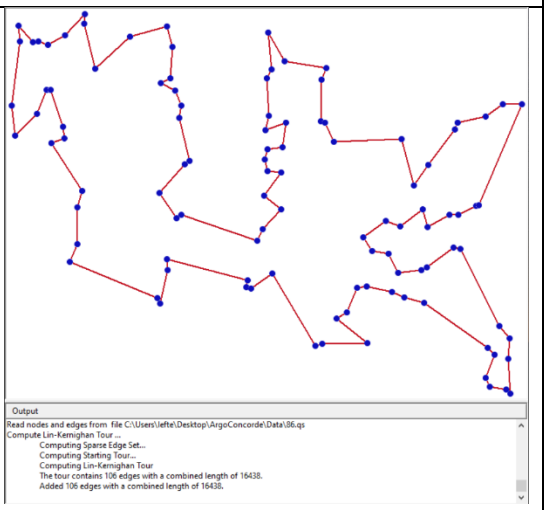
Συστάδα 82



Συστάδα 83



Συστάδα 84

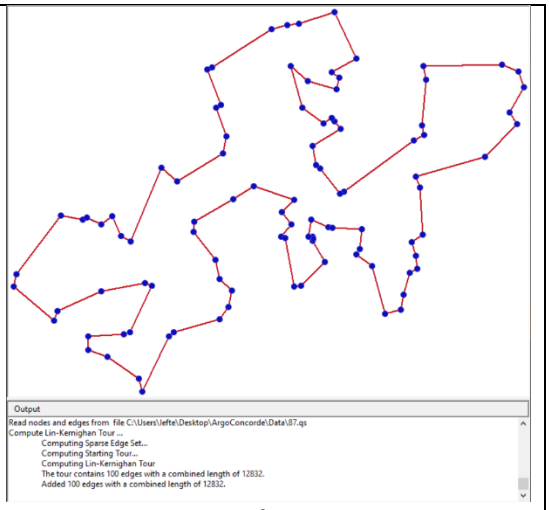


Συστάδα 85

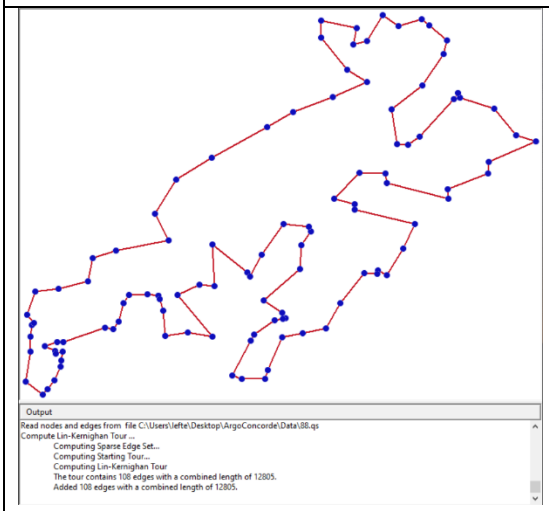
Προβολή αποτελεσμάτων συστάδων

Από το λογισμικό Concorde

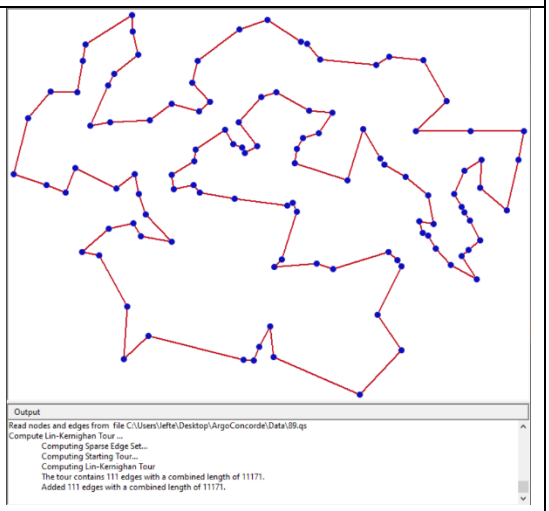
Σειρά: 86/87/88/89



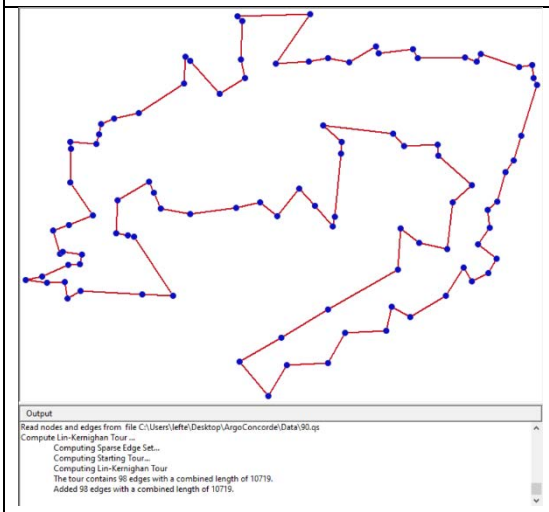
Συστάδα 86



Συστάδα 87



Συστάδα 88



Συστάδα 89

5.3.4 Σύνοψη αποτελεσμάτων υπερσυστάδων αργής επίλυσης

Στον παρακάτω πίνακα παίρνουμε συγκεντρωτικά όλα τα αποτελέσματα που έδωσε ο Concorde TSP Solver και το χρόνο που απαιτεί η επιθεώρηση τους κατά την αργή επίλυση:

Πίνακας 5.3 Αποτελέσματα υπερσυστάδων αργής επίλυσης

Συστάδα	Στάσεις	Απόσταση	Χρόνος
1	58	22503	8.684233
2	58	20803	8.174233
3	79	18559	8.201033
4	57	16949	6.9847
5	94	16971	8.224633
6	72	18394	7.9182
7	58	24173	9.185233
8	80	20324	8.763867
9	93	14618	7.4854
10	84	22178	9.4534
11	57	23440	8.932
12	64	22956	9.020133
13	75	18723	8.1169
14	87	18948	8.5844
15	92	18634	8.656867
16	68	20516	8.421467
17	63	17849	7.4547
18	87	19683	8.8049
19	99	18714	8.9142
20	71	12162	6.015267
21	82	12603	6.514233
22	67	12503	5.984233
23	86	17763	8.195567
24	86	15402	7.487267
25	73	18062	7.851933
26	87	18351	8.4053
27	78	16399	7.5197
28	66	13451	6.2353
29	74	14862	6.925267
30	92	13749	7.191367
31	72	16117	7.2351
32	78	11829	6.1487
33	70	15242	6.905933
34	67	14930	6.712333
35	75	12095	6.1285
36	91	15110	7.566333

37	88	12983	6.828233
38	95	17544	8.429867
39	85	17366	8.043133
40	100	13460	7.371333
41	92	14456	7.403467
42	81	13755	6.8265
43	82	14029	6.942033
44	75	13594	6.5782
45	70	11808	5.875733
46	112	16823	8.780233
47	92	17791	8.403967
48	111	16724	8.7172
49	105	16026	8.3078
50	114	16220	8.666
51	99	13901	7.4703
52	101	13420	7.392667
53	86	12734	6.686867
54	88	12127	6.571433
55	88	14032	7.142933
56	82	11509	6.186033
57	96	14047	7.4141
58	91	13233	7.003233
59	107	16854	8.622867
60	101	15287	7.952767
61	85	14011	7.036633
62	116	13972	8.058267
63	105	13441	7.5323
64	113	14762	8.195267
65	96	13013	7.1039
66	103	13136	7.374133
67	104	14219	7.732367
68	95	13136	7.107467
69	84	13982	6.9946
70	95	14639	7.558367
71	97	13890	7.400333
72	91	13450	7.068333
73	119	15502	8.617267
74	98	13863	7.425567
75	96	13887	7.3661
76	114	14921	8.2763
77	111	15855	8.4565
78	109	15492	8.280933
79	101	16959	8.454367
80	102	19627	9.2881
81	100	17903	8.704233
82	113	15517	8.421767

83	107	15223	8.133567
84	104	15119	8.002367
85	106	16438	8.464733
86	100	12832	7.182933
87	108	12805	7.4415
88	111	11171	7.0513
89	98	10719	6.482367

6. Συμπεράσματα και προτάσεις για περαιτέρω έρευνα

Σε αυτό το κεφάλαιο θα συγκρίνουμε τα αποτελέσματα που πήραμε από τις δυο διαφορετικές μεθόδους που χρησιμοποιήσαμε για να υποδιαιρέσουμε το δίκτυο μας σε συστάδες ώστε να είναι εφικτή η επιθεώρηση του.

6.1 Μηχανική και επιλεκτική συσταδιοποίηση για την γρήγορη επίλυση

Κατά την πρώτη μέθοδο που μελετήθηκε στο 4^{ο} κεφάλαιο , υποδιαιρούσαμε το δίκτυο συνεχόμενα σε όλο και περισσότερες συστάδες με χρήση του αλγορίθμου k-means clustering. Ωστόσο για να αποφύγουμε την δημιουργία πολύ μικρών συστάδων λόγω της ανομοιόμορφης κατανομής των στάσεων στον χώρο και τον τρόπο που λειτουργεί ο αλγόριθμος, διαλέξαμε να κάνουμε την συσταδιοποίηση σε επίπεδα και σε κάθε επόμενο βήμα συσταδιοποίησης να μην υποδιαιρείται η κάθε συστάδα σε άνω των 5 υποσυστάδων. Τελικώς πήραμε το ακολουθώ δέντρο από τις διαδοχικές συσταδιοποιήσεις.

Πίνακας 6.1 Απεικόνιση επιπέδων μηχανικής συσταδιοποίησης

Επίπεδο	*1*	*2*	*3*
Συστάδες	1	1.1	2.1.1
		1.2	2.1.2
		1.3	3.1.1
	2	2.1	3.1.2
		2.2	3.1.3
		2.3	3.3.1
		2.4	3.3.2
		2.5	3.4.1

3	3.1	3.4.2
	3.2	3.5.1
	3.3	3.5.2
	3.4	3.5.3
	3.5	4.1.1
4	4.1	4.1.2
	4.2	4.1.3
	4.3	4.1.4
	4.4	4.2.1
	4.5	4.2.2
5	5.1	4.2.3
	5.2	4.2.4
	5.3	4.3.1
	5.4	4.3.2
	5.5	4.3.3
		4.4.1
		4.4.2
		4.4.3
		4.5.1
		4.5.2
		5.1.1
		5.1.2
		5.3.1
		5.3.2
		5.5.1
		5.5.2

Θα αναφερόμαστε σε αυτή την διαδικασία συσταδιοποίηση ως μηχανική συσταδιοποίηση.

Ωστόσο όπως είδαμε στο 5ο κεφάλαιο ολοκληρώσαμε και μια διαδικασία συσταδιοποίησης, που με την εμπειρία που πήραμε από τα αποτελέσματα του 4ου κεφαλαίου και την χρήση αλγορίθμων για βοήθεια μας επιλέξαμε μονοί μας τις συστάδες. Θα αναφερόμαστε σε αυτή την διαδικασία ως επιλεκτική συσταδιοποίηση.

Το πρώτο αποτέλεσμα που μας ενδιαφέρει είναι το εύρος των τιμών που παίρνει ο χρόνος που απαιτείται για την επιθεώρηση τους.

Πίνακας 6.2 Σύγκριση αποτελεσμάτων μηχανικής και επιλεκτικής συσταδιοποίησης για γρήγορη επίλυση

Χρόνοι	**Μηχανική**	**Επιλεκτική**			
Ελάχιστος	1.696767	6.318166667	Ποσοστό Εντός(%)		Βελτίωση
Μέγιστος	11.63617	9.613033333	Μηχανική	Επιλεκτική	
4-12 ώρες	43	45	97.72727	100	1.02
5-11 ώρες	37	45	84.09091	100	1.19
6-10 ώρες	33	45	75	100	1.33
7-9 ώρες	17	30	38.63636	66.66667	1.73

7,5 - 8,5 ώρες	9	14	20.45455	31.11111	1.52

Βλέπουμε λοιπόν ότι με την επιλεκτική συσταδιοποίηση καταφέραμε πολύ μικρότερη διακύμανση των χρονών, σημαντικό το γεγονός ότι όλες μας οι συστάδες είναι εντός 2 ωρών από το επιθυμητό 8ωρο και δεν εμφανίζονται ακραίες τιμές που δεν βγάζουν νόημα.

Έπειτα μας ενδιαφέρει ο συνολικός χρόνος που απαιτείται και ο μέσος ορός χρόνου αλλά και η συνολική απόσταση που θα διανθεί τελικά όπως και ο μέσος ορός της απόστασης.

Πίνακας 6.3 Σύγκριση Χρόνων μηχανικής και επιλεκτικής συσταδιοποίησης για γρήγορη επίλυση

	Μηχανική	**Επιλεκτική**	**Βελτίωση**
Συνολικός Χρόνος	342.14965	337.7409	-4.4088
ΜΟ Χρόνων	7.776128409	7.505352	-0.2707762
Συνολική Απόσταση	1396331	1366939	-29392
ΜΟ Απόστασης	31734.79545	30376.42	-1358.3732

Όπως βλέπουμε με την επιλεκτική συσταδιοποίηση καταφέραμε βελτίωση και σε αυτούς τους τομείς καθώς το αρνητικό πρόσημο σημαίνει πως χρειαζόμαστε λιγότερη ώρα και πρέπει να διανύσουμε μικρότερη απόσταση για την ολοκλήρωση της επιθεώρησης.

Μοναδικό μειονέκτημα αποτελεί ότι στην επιλεκτική συσταδιοποίηση πρέπει να επιθεωρήσουμε 45 συστάδες ενώ στην μηχανική 44. Το οποίο θα μπορούσε να βελτιωθεί με ακόμα πιο προσεχτική επιλογή συστάδων.

6.2 Μηχανική και επιλεκτική συσταδιοποίηση για την αργή επίλυση

Για την αργή επίλυση στην μηχανική συσταδιοποίηση του δικτύου κάναμε χρήση 4 επίπεδων. Όπως είδαμε και στο 4^{ο} κεφάλαιο χρησιμοποιήσαμε τα 3 ήδη υπάρχοντα επίπεδα και προσθέσαμε ακόμα με κριτήριο τον χρόνο.

Για το εύρος των τιμών του χρόνου έχουμε:

Πίνακας 6.4 Σύγκριση αποτελεσμάτων μηχανικής και επιλεκτικής συσταδιοποίησης για αργή επίλυση

Χρόνοι	**Μηχανική**	**Επιλεκτική**			
Ελάχιστος	3.128267	5.875733	Ποσοστό Εντος(%)		
Μέγιστος	11.82117	9.4534	Μηχανική	Επιλεκτική	Βελτίωση
4-12 ώρες	86	89	96.62921	100	1.03
5-11 ώρες	78	89	87.64045	100	1.14
6-10 ώρες	68	87	76.40449	97.75281	1.28
7-9 ώρες	37	65	41.57303	73.03371	1.76
7,5 - 8,5 ώρες	21	29	23.59551	32.58427	1.38

Παρατηρούμε και εδώ αρκετά καλή βελτίωση στην διασπορά των τιμών του χρόνου και κυρίως το γεγονός ότι η συντριπτική πλειονότητα των συστάδων μπορούν να επιτηρηθούν από 6 έως 10 ώρες (ποσοστό 97,75%)

Συνολικά για χρόνο και απόσταση που απαιτεί η επιθεώρηση του δικτύου αλλά και για πόσο αναλογεί κατά μέσο ορό σε κάθε συστάδα έχουμε:

Πίνακας 6.5 Σύγκριση Χρόνων μηχανικής και επιλεκτικής συσταδιοποίησης για αργή επίλυση

	Μηχανική	**Επιλεκτικη**	**Βελτίωση**
Συνολικός Χρόνος	697.8107	683.8316	-13.9791
ΜΟ Χρόνων	15.85933	15.19626	-0.66308
Συνολική Απόσταση	1441369	1394772	-46597
ΜΟ Απόστασης	32758.39	30994.93	-1763.45

Όπου βλέπουμε και ότι εδώ πέτυχαμε μια αρκετά καλή βελτίωση. Κάθε συστάδα κατά μέσο ορό θα χρειαστεί 40 λεπτά λιγότερα να επιθεωρηθεί και ο οδηγός θα πρέπει να διανύσει 1,76 χιλιόμετρα λιγότερο.

Παρατηρούμε εδώ ότι ο τελικός αριθμός συστάδων είναι όμοιος, 89 συστάδες και στις δυο περιπτώσεις οπότε η σύγκριση μας για βελτίωση μας δίνει ακόμα καλύτερη αντίληψη για τα περιθώρια που υπάρχουν.

6.3 Προτάσεις για περαιτέρω ερευνά

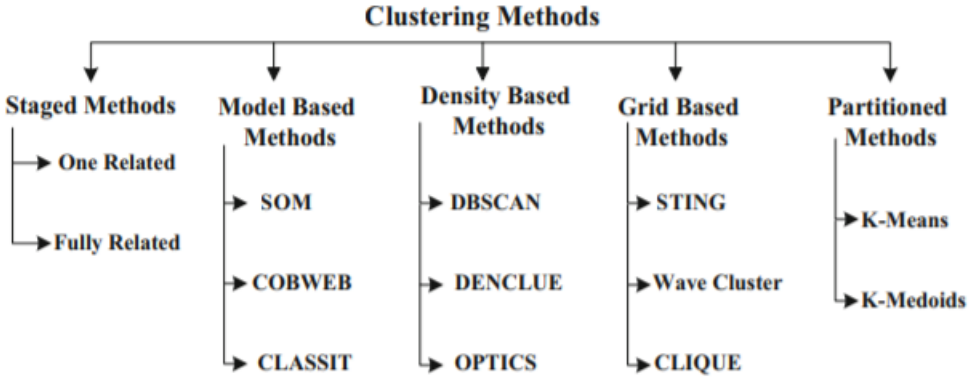
Όπως είδαμε και στο 2ο κεφάλαιο η ερευνά για ένα τέτοιου είδους πρόβλημα είναι άπειρη. Πέρα από το πλήθος των αλγορίθμων που έχουν εφευρεθεί ανά τα χρονιά για την επίλυση του TSP, των ευρετικών και μεθευρετικών μεθόδων βελτίωσης υπάρχει και μεγάλο πλήθος αλγορίθμων για την διαδικασία της συσταδιοποίησης, οπότε ο συνδυασμός τους για την αντιμετώπιση ενός τέτοιου προβλήματος αφήνει πάρα πολλές επιλογές. Το ερώτημα είναι πάντα πως θα πάρουμε όσο το δυνατόν πιο καλά αποτελέσματα μπορούμε σε όσο το δυνατόν πιο σύντομο χρόνο διότι η αναλυτική λύση δεν φαίνεται ακόμα με τα υπολογιστικά μέσα που διαθέτουμε να μπορεί να δοθεί, για την ακρίβεια φαίνεται να είμαστε αρκετά μακριά ακόμα.

Οπότε σαν πρώτη ιδέα για περαιτέρω ερευνά θα ήταν ο συνδυασμός μιας διαφορετικής μεθόδου εύρεσης βέλτιστης διαδρομής με έναν διαφορετικό αλγόριθμο συσταδιοποίησης και η παρατήρηση σε πόσο καλά αποτελέσματα μπορούμε να έρθουμε.

Μερικοί από τους ακριβείς και ευρετίκους αλγορίθμους που μπορούν να χρησιμοποιηθούν για την επίλυση του TSP είναι οι:

- **Branch and Bound**
- **Αλγόριθμος Cutting Plane**
- **Αλγόριθμος Lin-Kernighan**
- **Πλησιέστερος Γείτονας**
- **Γενετικός Αλγόριθμος**
- **Άπληστος Αλγόριθμος**
- **Ευρετικός Αλγόριθμος Χριστοφίδη**
- **Αναζήτηση Ταμπού**
- **Βελτιστοποίηση Αποικίας Μυρμηγκιών (Ant Colony Optimization – ACO)**
- **Βελτιστοποίηση Σμήνους Σωματιδίων (Particle Swarm Optimization – PSO)**
- **Προσομοιωμένη Ανόπτηση (Simulated Annealing – SA)**

Σε συνδυασμό με μερικούς από τους παρακάτω αλγορίθμους συσταδιοποίησης:



Στην βιβλιογραφία και ειδικά γύρω από τις παραλλαγές **vehicle routing problem with hard time windows (VRPHTW), capacitated vehicle routing problem (CVRP), clustered vehicle routing problem (CluVRP), general traveling salesman problem (GTSP) και clustered travelling salesman problem (CTSP)** κάποιος μπορεί να βρει αναφορές σε τέτοιου είδους τεχνικές και προσπάθειες βελτιστοποίησης.

Τέλος παρατηρούμε ότι τα τελευταία χρονιά στο επιστημονικό μέτωπο χρησιμοποιείται ολοένα και περισσότερο η μηχανική μάθηση. Μια μελέτη που θα μπορούσε να γίνει πάνω σε αυτό το πρόβλημα με την χρήση της μηχανικής μάθησης θα ήταν η εξής:

- Δεδομένου ενός πλήθους σημείων και συγκεκριμένης διάταξης τους στον χώρο, ανάλογα το πλήθος και τις ιδιότητες μια εκ των ελάχιστων γεωμετριών που τα καλύπτει(κύκλος , ορθογώνιο παραλληλόγραμμο, convex hull) ποια προβλέπουμε ότι θα είναι μια αρκετά βέλτιστη διαδρομή (λύση περίπου 10% πάνω από το κάτω όριο Held-Karp) ;

Δηλαδή να εκπαιδεύσουμε έναν αλγόριθμο με την παραγωγή τυχαίων σημείων στα όρια ενός γράφου, έπειτα να βρίσκει το πλήθος τους, τον ελάχιστο περιβάλλοντα κύκλο, έπειτα να χρησιμοποιεί μια αρκετά καλή και αποδοτική ευρετικη μέθοδο από αυτές που έχουν εφευρεθεί (π.χ. Lin – Kernighan) και τέλος να κατατάσσει τα αποτελέσματα ώστε μετά να μπορεί να κάνει μια πρόβλεψη για το περίπου πόσο θα είναι η τελική βέλτιστη διαδρομή.

Και μετά βάση αυτού του μοντέλου και τις απαιτήσεις μας να δημιουργήσουμε έναν αλγόριθμο συσταδιοποίησης όμοιο της στρατηγικής που ακολουθήσαμε κατά την επιλεκτική επίλυση. Αλγοριθμικά θα μπορούσε να εκφραστεί ως εξής:

- Από το σύνολο των σημείων που έχουμε τοποθετημένα στον γράφο μας διαλέγουμε τα τέσσερα σημεία που βρίσκονται πιο κοντά στις γωνίες του γράφου (ένα για κάθε γωνιά).
- Δημιουργούμε για κάθε σημείο μια συστάδα προσθέτοντας γειτονικά σημεία μέχρι η πρόβλεψη του αλγορίθμου της μηχανικής μάθησης για το πόση ώρα θα χρειαστεί η επιθεώρηση τους να μας δώσει το επιθυμητό μας νούμερο.

- Αφού ολοκληρωθεί η δημιουργία μιας συστάδας αφαιρούμε τα σημεία της από τον γράφο
- Επαναλαμβάνουμε τα βήματα 2 και 3 μέχρι να μην υπάρχει άλλο σημείο στον γράφο.

7. Βιβλιογραφία

1. Καρλαύτης Μ., Λαγαρός Ν. (2010). Επιχειρησιακή έρευνα και βελτιστοποίηση για μηχανικούς. Εκδόσεις Συμμετρία.
2. Lawler, E. L. (1985). The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization (Repr. with corrections. ed.). John Wiley & sons. ISBN 978-0471904137.
3. Velednitsky, Mark (2017). "Short combinatorial proof that the DFJ polytope is contained in the MTZ polytope for the Asymmetric Traveling Salesman Problem". Operations Research Letters.
4. TolgaBektas. The multiple traveling salesman problem: an overview of formulations and solution procedures.
5. Toth, P.; Vigo, D., eds. (2002). The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications. 9. Philadelphia: Society for Industrial and Applied Mathematics. ISBN 0-89871-579-2.
6. Alvina G.H.Kek, Ruey Long Cheu, Qiang Meng. Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots
7. J.-F. Cordeau, Guy Desaulniers, Jacques Desrosiers, Marius M. Solomon, Fran¸cois Soumis, The VRP with Time Windows.
8. Feiyue Li, Bruce Golden, Edward Wasil. The open vehicle routing problem: Algorithms, large-scale test problems, and computational results.
9. Claudio Contardo, Rafael Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints.
10. L. Dongmei, W. Xiangbin and W. Dong, "Exact Heuristic Algorithm for Traveling Salesman Problem," 9th International Conference for Young Computer Scientists, 2008.
11. J. Štencek, Traveling salesman problem, Jyväskylä: JAMK University of Applied Sciences, 2013

12. H. A. Abdulkarim and I. F. Alshammari, "Comparison of Algorithms for Solving Traveling Salesman Problem," International Journal of Engineering and Advanced Technology (IJEAT), 2015.
13. K. Helsgaun, "An effective implementation of the Lin-Kernighan traveling salesman heuristic," European Journal of Operational Research, 2000.
14. Blum, C.; Roli, A. (2003). "Metaheuristics in combinatorial optimization: Overview and conceptual comparison".
15. A. Khushboo, A. Samiksha and T. Rohit, "Solving TSP using Genetic Algorithm and Nearest Neighbour Algorithm and their Comparison," International Journal of Scientific & Engineering Research, 2016.
16. N. Kumar, Karambir and R. Kumar, "A genetic algorithm approach to study travelling salesman problem," Journal of Global Research in Computer Science, 2012.
17. Helena R. Lourenço,Olivier C. Martin, Thomas Stützle. Iterated Local Search.
18. Ιωάννης Μαρινάκης, Μαγδαληνή Μαρινάκη, Νικόλαος Φ.Ματσατσίνης, ΚΚωνσταντίνος Ζοπουνίδης. Μεθευρετικοι και Εεξελικτικοί Ααλγόριθμοι σε προβληματα διοικητικης επιστημης.
19. G. A. Croes, A method for solving traveling salesman problems. Operations Res. 6 (1958).
20. Blazinskas, Andrius; Misevicius, Alfonsas (2011). "Combining 2-OPT, 3-OPT and 4-OPT with K-SWAP-KICK perturbations for the traveling salesman problem".
21. S. Lin and B. W. Kernighan, "An Effective Heuristic Algorithm for the TravelingSalesman Problem," Operations Research.
22. Hartigan, J. A.; Wong, M. A. (1979). "Algorithm AS 136: A k-Means Clustering Algorithm". Journal of the Royal Statistical Society, Series C.
23. Junjie Wu, Advances in K-means Clustering: a Data Mining Thinking, 2012
24. Joseph O'Rourke (1985), "Finding minimal enclosing boxes", Parallel Programming.
25. Κεραμιώτης Κ. (2006). Παραμετρική Μελέτη Γενετικού Αλγόριθμου για την Επίλυση του Προβλήματος Ομαδικού Προσανατολισμού. Διπλωματική Εργασία, Πανεπιστήμιο Αιγαίου, Σχολή Επιστημών της Διοίκησης, Τμήμα Μηχανικών Οικονομίας και Διοίκησης.
26. Κινικλής Β. (2009). Αλγόριθμος επαναληπτικής επανασύνδεσης διαδρομών με χρήση τοπικής αναζήτησης για το πρόβλημα δρομολόγησης οχημάτων. Διπλωματική Εργασία, Πολυτεχνείο Κρήτης, Τμήμα Μηχανικών Παραγωγής και Διοίκησης.
27. Λαζαρόπουλος Ν. (2010). Βέλτιστη δρομολόγηση οχήματος σε συνθήκες αβεβαιότητας. Διπλωματική Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.
28. Λιγνός Γ. (2010). Επίλυση Προβλημάτων Δρομολόγησης Οχημάτων με Χρονικά Παράθυρα και Χωρητικότητα (VRPTW). Διπλωματική Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Χημικών Μηχανικών.

29. ΑΛΕΞΙΟΣ ΤΡΥΦΩΝΑΣ ΧΑΡΑΛΑΜΠΟΥΣ (2019). Το Πρόβλημα του Περιοδεύοντος Πωλητή: Ανάλυση Ερευνητικού Πεδίου, Αλγόριθμοι Επίλυσης και Επιχειρησιακές Εφαρμογές. ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
30. Νικόλαος Σ. Στυλιανού(2013) Προσεγγίζοντας το Πρόβλημα του Πλανόδιου Πωλητή ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
31. Θεολογία Μ. Μουστάκα(2014). ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΔΙΑΝΟΜΗΣ ΑΓΑΘΩΝ. ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
32. Comparison of TSP Algorithms Project for Models in Facilities Planning and Materials Handling December 1998 Participants: Byung-In Kim Jae-Ik Shim Min Zhang
33. Heuristics for the Traveling Salesman Problem. Christian Nilsson. Linkoping University
34. Tolga Bektas, (2005). The multiple Traveling Salesman Problem: an overview of fomulations and solutions procedures, Elsevier.
35. Rosenkrantz D. J, Stearns R. E, Lewis P. M. (1977), An analysis of several heuristics for the traveling salesman problem.

Online σύνδεσμοι:

36. http://www.math.uwaterloo.ca/tsp/concorde/index.html
37. https://docs.qgis.org/3.16/en/docs/user_manual/index.html