



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**  
**ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**

**Εξόρυξη τροχιών κινούμενων αντικειμένων**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

της

**ΑΛΕΞΑΝΔΡΑΣ Γ. ΒΛΑΧΑΚΗ**

**Επιβλέπων :** Ιωάννης Βασιλείου  
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2013





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

## Εξόρυξη τροχιών κινούμενων αντικειμένων

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

**ΑΛΕΞΑΝΔΡΑΣ Γ. ΒΛΑΧΑΚΗ**

**Επιβλέπων :** Ιωάννης Βασιλείου  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 8<sup>η</sup> Ιουλίου 2013.

.....  
Ιωάννης Βασιλείου  
Καθηγητής Ε.Μ.Π.

.....  
Κώστας Κοντογιάννης  
Καθηγητής Ε.Μ.Π.

.....  
Θοδωρής Δαλαμάγκας  
Ερευνητής Β' ΙΠΣΥ/Ε.Κ. "Αθηνά"

Αθήνα, Ιούλιος 2013

.....

**ΑΛΕΞΑΝΔΡΑ Γ. ΒΛΑΧΑΚΗ**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © 2013 Αλεξάνδρα Γ. Βλαχάκη

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τους καθηγητές κ. Ιωάννη Βασιλείου και κ. Τιμολέοντα Σελλή για την ευκαιρία που μου έδωσαν να συνεργασθώ με το ΙΠΣΥ/Ε.Κ. «Αθηνά». Επίσης, θα ήθελα να ευχαριστήσω το Δημήτρη Σκούτα και το Γιώργο Σκούμα από το ΙΠΣΥ/Ε.Κ. «Αθηνά» για την πολύτιμη βοήθειά τους στην εκπόνηση της παρούσας διπλωματικής εργασίας. Τέλος, θα ήθελα να ευχαριστήσω θερμά την οικογένειά μου και τους φίλους μου για την αμέριστη συμπαράστασή τους και τη στήριξη της προσπάθειάς μου.

Αυτή η διπλωματική εργασία αφιερώνεται στο φίλο μου, Παύλο, ο οποίος ήταν πάντα δίπλα μου και χωρίς τον οποίο δε θα είχα καταφέρει τόσα πολλά.

Αλεξάνδρα Γ. Βλαχάκη

Αθήνα, Ιούλιος 2013



## Περίληψη

Τα τελευταία χρόνια η ανάπτυξη τεχνολογιών GPS, RFID, τηλεμετρίας και τηλεπικοινωνιών έχει οδηγήσει σε μαζική διάδοση δεδομένων που καταγράφουν τις γεωγραφικές θέσεις κινούμενων αντικειμένων σε διάφορες χρονικές στιγμές. Η δυνατότητα εξαγωγής χρήσιμης γνώσης από τον τεράστιο όγκο δεδομένων που έχει συλλεχθεί έχει προκαλέσει έντονο ερευνητικό ενδιαφέρον για την ανάλυση των δεδομένων αυτών με αυτόματα ή ημιαυτόματα μέσα και την ανακάλυψη ενδιαφερόντων προτύπων και συσχετίσεων.

Ο σκοπός της διπλωματικής εργασίας είναι η εύρεση αντικειμένων που κινούνται σε κοντινή απόσταση από ένα δοθέν αντικείμενο για κάποιο συγκεκριμένο χρονικό διάστημα. Για την επίλυση του προβλήματος, δίνεται ιδιαίτερη έμφαση στη χρήση κατάλληλου μέτρου ομοιότητας των τροχιών, που θα αποτυπώνει με όσο το δυνατόν μεγαλύτερη ακρίβεια τη διαφορά στις κινήσεις των αντικείμενων.

Η ανάπτυξη μεθοδολογίας για την εύρεση των κ κοντινότερων κινούμενων γειτόνων στοχεύει στην αποδοτική επίλυση του προβλήματος ως προς το χρόνο εκτέλεσης, καθώς και ως προς την αξιοπιστία των αποτελεσμάτων, δηλαδή την εμφάνιση αποτελεσμάτων που είναι διαισθητικά αναμενόμενα.

**Λέξεις Κλειδιά:** κινούμενα αντικείμενα, τροχιά, χωρο-χρονική ομοιότητα, κ κοντινότεροι γείτονες





## **Abstract**

Recently the development of GPS, RFID and telemetry technologies has enabled a massive distribution of trajectory data. The possibility to extract potentially useful knowledge from this huge amount of data collected has initiated considerable research studies in order to analyze these data in an automatic or semi-automatic way and discover interesting patterns and correlations.

The goal of this diploma thesis is the discovery of objects that move close together with a given query object within a time period. To solve the problem, particular emphasis is given on using an appropriate trajectory similarity measure that will accurately capture the differences in the objects' movements.

The methodology developed to find the  $k$  nearest moving neighbors aims at solving the problem efficiently with respect to the execution time, as well as the reliability of the results, i.e. the output of results that were intuitively expected.

**Keywords:** moving objects, trajectory, spatio-temporal similarity,  $k$  nearest neighbors



## Πίνακας περιεχομένων

1	Εισαγωγή.....	1
1.1	Εξόρυξη γνώσης από βάσεις κινούμενων αντικειμένων.....	1
1.2	Αντικείμενο διπλωματικής.....	3
1.2.1	Συνεισφορά.....	3
1.3	Οργάνωση κειμένου.....	4
2	Σχετικές εργασίες.....	5
2.1	Ανάλυση ομοιότητας τροχιών.....	5
2.2	Απλοποίηση τροχιών.....	10
2.3	Εύρεση ομάδων κινούμενων αντικειμένων.....	13
2.4	Εύρεση k-NN κινούμενων αντικειμένων.....	16
2.5	k-NN σε προβλήματα εξόρυξης γνώσης από τροχιές.....	17
3	Ορισμός προβλήματος.....	19
3.1	Προαπαιτούμενα.....	19
3.2	Το πρόβλημα εύρεσης κ κοντινότερων γειτόνων σε ιστορικές βάσεις δεδομένων.....	20
4	Μέθοδοι επίλυσης του προβλήματος.....	23
4.1	Προεπεξεργασία δεδομένων.....	23
4.2	Εύρεση των κ κοντινότερων γειτόνων.....	25
5	Πειραματική ανάλυση και αξιολόγηση.....	37
5.1	Σύνολα δεδομένων.....	37
5.2	Περιβάλλον εκτέλεσης και κριτήρια αξιολόγησης.....	40
5.3	Αποτελέσματα.....	40
6	Επίλογος.....	51
6.1	Σύνοψη και συμπεράσματα.....	51
6.2	Μελλοντικές επεκτάσεις.....	51
7	Βιβλιογραφία.....	55



# 1

## Εισαγωγή

### 1.1 Εξόρυξη γνώσης από βάσεις κινούμενων αντικειμένων

Εξόρυξη δεδομένων (data mining) είναι η διαδικασία ανάλυσης μεγάλου όγκου δεδομένων με αυτόματα ή ημιαυτόματα μέσα για την εξαγωγή χρήσιμης και μέχρι πρότινος άγνωστης πληροφορίας [WFH11]. Το συνεχώς αυξανόμενο ενδιαφέρον για τη διαδικασία εξόρυξης δεδομένων πηγάζει από τον τεράστιο όγκο δεδομένων που έχουν συλλεχθεί και τη χρησιμότητα εξαγωγής γνώσης (πρότυπα, συσχετίσεις, κανόνες, περιορισμοί) από αυτά για πληθώρα εφαρμογών, όπως ανάλυση και διαχείριση αγοράς, εξόρυξη κειμένου κι ανάλυση Ιστού και λήψη αποφάσεων σε γεωγραφικά πληροφοριακά συστήματα.

Από το σύνολο των δεδομένων που έχουν συλλεχθεί εκτιμάται ότι περισσότερο από 80% περιλαμβάνουν γεωχωρική πληροφορία [F92, O90]. Ιδιαίτερα τα τελευταία χρόνια παρατηρείται μαζική διάδοση γεωχωρικών δεδομένων κινούμενων χρηστών με τη βοήθεια διαφορετικών τεχνολογιών:

- κινούμενα αντικείμενα εξοπλισμένα με GPS, όπως αυτοκίνητα, άνθρωποι και ζώα
- τεχνικές εντοπισμού θέσης από το «διαδίκτυο των πραγμάτων» (Internet of Things) [C10], όπου συσκευές συνδέονται ασύρματα στο διαδίκτυο με τεχνολογίες RFID, δικτύων αισθητήρων (sensor networks) κ.ά.
- συλλογή πληροφορίας από έξυπνες συσκευές
  - θέσεις χρηστών από υπηρεσίες check-in κοινωνικών δικτύων, όπως το Foursquare<sup>1</sup> και το Facebook<sup>2</sup>

- φωτογραφίες με πληροφορία γεωγραφικής θέσης (geotagged photos) από εφαρμογές, όπως το Flickr<sup>3</sup> και το Picasa<sup>4</sup>
- διαδρομές χρηστών, οι οποίοι θέλουν να τις μοιραστούν με άλλους χρήστες (route-sharing) με εφαρμογές όπως το OpenStreetMap<sup>5</sup>, το EveryTrail<sup>6</sup>, το Bikely<sup>7</sup> και το ShareMyRoutes<sup>8</sup>

Αναλύοντας τα δεδομένα κίνησης των χρηστών μπορούν να ανακαλυφθούν ενδιαφέροντα πρότυπα κίνησης, να κατανοηθούν και να προβλεφθούν συμπεριφορές, καθώς και να αναπτυχθούν νέες εφαρμογές και υπηρεσίες. Ανάλογα με το σκοπό κάθε εφαρμογής, η εξόρυξη γνώσης μπορεί να γίνει από βάση δεδομένων καταγραφής (tracking database) ή από ιστορική βάση δεδομένων (historical database) [GBX10].

Η βάση δεδομένων καταγραφής αποθηκεύει συνεχώς τις τρέχουσες θέσεις κινούμενων αντικειμένων και επιτρέπει ερωτήματα για τις τρέχουσες και τις αναμενόμενες θέσεις στο κοντινό μέλλον. Με αυτή την προσέγγιση, για παράδειγμα, μια εταιρία ταξί μπορεί να βρει το κοντινότερο ταξί σε επιβάτη που ζητά μεταφορά.

Αντίθετα, η ιστορική βάση δεδομένων αποθηκεύει όλο το ιστορικό της κίνησης, όπως τις διαδρομές οχημάτων μιας εταιρείας logistics για μια ημέρα, ένα μήνα ή ένα χρόνο, κι επιτρέπει σύνθετη ανάλυση των τροχιών για κατηγοριοποίηση (classification), ομαδοποίηση (clustering) ή εύρεση ενδιαφερόντων προτύπων τροχιών (trajectory patterns) με βάση τη χωρική ή τη χωρο-χρονική ομοιότητά τους. Η ανάλυση των τροχιών στο παρελθόν μπορεί να συμβάλλει σε εφαρμογές βελτιστοποίησης μεταφοράς, όπως στη βελτίωση της δρομολόγησης παραδόσεων για μια εταιρεία logistics ή στην αποδοτικότερη ομαδοποίηση επιβατών για carpooling. Επίσης, επιτρέπει στους οικολόγους να αναλύσουν τη συμπεριφορά ζώων μελετώντας τις κινήσεις τους στο παρελθόν.

---

<sup>1</sup><https://foursquare.com/>

<sup>2</sup><https://www.facebook.com/>

<sup>3</sup><http://www.flickr.com/>

<sup>4</sup><http://picasa.google.com/>

<sup>5</sup><http://www.openstreetmap.org/>

<sup>6</sup><http://www.everytrail.com/>

<sup>7</sup><http://www.bikely.com/>

<sup>8</sup><http://www.sharemyroutes.com/>

## 1.2 Αντικείμενο διπλωματικής

Η παρούσα διπλωματική εργασία μελετά την εύρεση αντικειμένων που κινούνται μαζί στο χώρο και στο χρόνο. Στη βιβλιογραφία αναζητούνται συνήθως ομάδες (clusters) κινούμενων αντικειμένων, όπως flocks [GK08, VBT09], convoys [JSZ08, JYZ08], moving clusters [KNB05] και swarms [LDH+10], τα οποία παραμένουν σε κοντινή μεταξύ τους απόσταση (σε σχέση με ένα δοθέν κατώφλι) για επαρκώς μεγάλα χρονικά διαστήματα, *διαδοχικά ή μη*.

Σε αυτή την εργασία θα ακολουθήσουμε μια διαφορετική προσέγγιση. Συγκεκριμένα, αναζητάμε αντικείμενα που κινούνται σε κοντινή απόσταση από ένα δοθέν αντικείμενο για κάποιο συγκεκριμένο χρονικό διάστημα. Το δοθέν αντικείμενο μπορεί να είναι ένα πραγματικό αντικείμενο σε μια ιστορική βάση δεδομένων ή ένα εικονικό που αναπαριστά μια επιθυμητή κίνηση. Μελετώντας τους κοντινότερους γείτονες πραγματικών αντικειμένων, μπορούμε να χαρακτηρίσουμε και να κατηγοριοποιήσουμε τη συμπεριφορά αυτών των αντικειμένων ή να κάνουμε προβλέψεις και προτάσεις (recommendations) για αυτά. Εισάγοντας εικονικές τροχιές, μπορούμε να βρούμε αντικείμενα των οποίων η κίνηση μοιάζει με ένα επιθυμητό πρότυπο.

Για την εύρεση των κοντινότερων γειτόνων εξετάζουμε πόσο όμοια είναι τα τμήματα τροχιάς που διένυσαν τα αντικείμενα κατά τα αντίστοιχα χρονικά διαστήματα. Είναι σημαντικό να συγκρίνονται ολόκληρα τμήματα, αντί για μεμονωμένα σημεία των τροχιών, ώστε να ληφθεί υπόψη και η κατεύθυνση. Επίσης, επειδή δεν είναι αποδοτική η εξαντλητική αναζήτηση για την εύρεση κοντινότερων γειτόνων, προτείνονται δύο μέθοδοι για να περιοριστεί ο χώρος αναζήτησης λύσεων. Η πρώτη μέθοδος αφορά στη χρήση ουράς προτεραιότητας, ώστε να κλαδεύονται οι τροχιές που απέχουν περισσότερο από τη δοθείσα σε σχέση με τις υπόλοιπες. Η δεύτερη μέθοδος προτείνει την προσεγγιστική αναπαράσταση των τροχιών με λιγότερα τμήματα, μειώνοντας έτσι τους υπολογισμούς αλλά διατηρώντας ικανοποιητική ακρίβεια αποτελεσμάτων.

### 1.2.1 Συνεισφορά

Η συνεισφορά της εργασίας συνίσταται στα εξής βασικά σημεία::

- Ακριβέστερη ανάλυση της χωρο-χρονικής ομοιότητας των τροχιών για την εύρεση των  $k$  κοντινότερων γειτόνων ( $k$  Nearest Neighbors,  $k$ -NN)
- Αποδοτική επίλυση του προβλήματος  $k$ -NN
- Δυνατότητα μείωσης της επιθυμητής ακρίβειας για ταχύτερα αποτελέσματα.

## **1.3 Οργάνωση κειμένου**

Τα κεφάλαια που ακολουθούν οργανώνονται ως εξής:

Στο κεφάλαιο 2 γίνεται μια σύντομη αναφορά σε εργασίες σχετικές με το αντικείμενο της διπλωματικής. Στο κεφάλαιο 3 περιγράφεται αναλυτικά το πρόβλημα που μελετάμε. Στο κεφάλαιο 4 αναλύονται μέθοδοι επίλυσης του προβλήματος και επεκτάσεις τους. Στο κεφάλαιο 5 παρουσιάζονται τα πειραματικά αποτελέσματα και οι μετρήσεις επίδοσης από την εκτέλεση των μεθόδων που αναπτύχθηκαν. Τέλος, στο κεφάλαιο 6 δίνονται τα συμπεράσματα από όλη την εργασία, καθώς και πιθανές μελλοντικές επεκτάσεις.



# 2

## Σχετικές εργασίες

Η εύρεση αντικειμένων που κινούνται σε κοντινές τροχιές βασίζεται στη σύγκριση της χωρο-χρονικής ομοιότητας των τροχιών τους και στην παρούσα εργασία εξετάζεται με τη μορφή συναθροιστικών ερωτημάτων  $k$  κοντινότερων γειτόνων (aggregate  $k$ -NN queries). Για την επιτάχυνση των αποτελεσμάτων η διαδικασία εύρεσης μπορεί να γίνει προσεγγιστικά σε απλοποιημένες τροχιές.

Στο κεφάλαιο αυτό περιγράφουμε υπάρχουσες εργασίες στη βιβλιογραφία σχετικές με την ανάλυση ομοιότητας τροχιών, την απλοποίηση τροχιών, την εύρεση αντικειμένων που κινούνται σε κοντινές τροχιές είτε με τη μορφή ομάδων είτε με τη μορφή  $k$ -NN, καθώς και τη χρήση  $k$ -NN σε διάφορα προβλήματα εξόρυξης γνώσης από τροχιές.

### 2.1 Ανάλυση ομοιότητας τροχιών

#### Ευκλείδεια Απόσταση

Μια απλή προσέγγιση για τον προσδιορισμό της ομοιότητας δύο τροχιών είναι ο υπολογισμός του αθροίσματος των ευκλείδειων αποστάσεων μεταξύ των αντίστοιχων θέσεων στις δύο τροχιές.

$$D_{\text{Euclidean}}(P_{1..n}, Q_{1..n}) = \sum_{t=1}^n \|P_t - Q_t\|,$$

όπου  $P, Q$  οι δυο τροχιές,

$P_t, Q_t$  η θέση στην τροχιά  $P, Q$  αντίστοιχα τη χρονική στιγμή  $t$   
και η ευκλείδεια απόσταση μεταξύ των δυο σημείων ορίζεται ως:

$$\|P_t - Q_t\| = \sqrt{(P_t, x - Q_t, x)^2 + (P_t, y - Q_t, y)^2}$$

Παρά τον εύκολο υπολογισμό της, η ευκλείδεια απόσταση δεν αποτελεί κατάλληλο μέτρο απόστασης τροχιών, καθώς δε λαμβάνει υπόψη εγγενή χαρακτηριστικά των τροχιών, όπως διεύθυνση, μήκος τροχιάς κ.ά.

### Απόσταση OWD (One Way Distance)

Μια διαφορετική προσέγγιση είναι η σύγκριση του σχήματος των τροχιών χρησιμοποιώντας το ολοκλήρωμα της απόστασης των σημείων της μιας τροχιάς από την άλλη [LS05].

$$D_{\text{OWD}}(T_1, T_2) = \frac{1}{|T_1|} \int_{p \in T_1} D_{\text{point}}(p, T_2) dp,$$

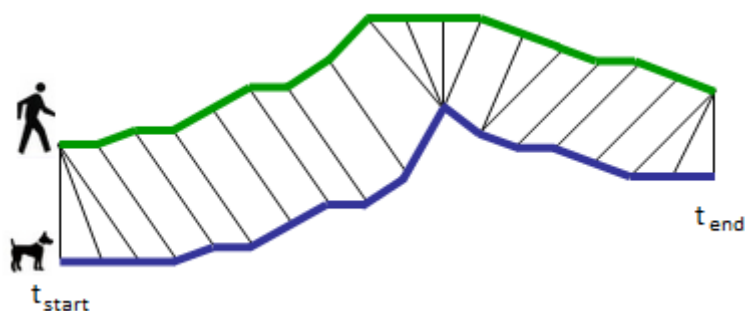
όπου η απόσταση σημείου  $p$  από τροχιά  $T_2$  ορίζεται ως:

$$D_{\text{point}}(p, T_2) = \min_{q \in T_2} D_{\text{euclidean}}(p, q).$$

Ωστόσο, η απόσταση OWD αγνοεί το χρονικό παράγοντα, αφού δε λαμβάνεται υπόψη η σχετική θέση των σημείων των δύο τροχιών τις αντίστοιχες χρονικές στιγμές.

### Απόσταση που βασίζεται σε ευθυγράμμιση

Στη βιβλιογραφία έχουν προταθεί επίσης αρκετά μέτρα απόστασης (Dynamic Time Warping, Edit Distance on Real sequence, Edit distance with Real Penalty, Longest Common Subsequence) που βασίζονται σε χρονική ευθυγράμμιση των τροχιών. Όπως απεικονίζεται στο Σχήμα 2.1, κατά την ευθυγράμμιση μπορεί να γίνει μεταβολή του χρόνου τοπικά σε μια τροχιά, αλλά όχι αναδιάταξη των σημείων.



Σχήμα 2.1: Παράδειγμα χρονικής ευθυγράμμισης τροχιάς δυο κινούμενων αντικειμένων

Η απόσταση Dynamic Time Warping (DTW), που χρησιμοποιείται ευρύτατα στη σύγκριση χρονοσειρών, στοιχίζει κατάλληλα την ακολουθία σημείων των δύο τροχιών, έτσι ώστε η συνολική απόσταση τους ως άθροισμα επιμέρους αποστάσεων να ελαχιστοποιείται [YJF98].

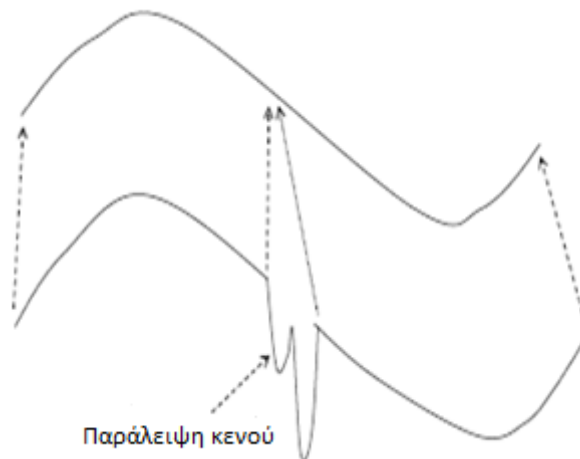
$$DTW(P_{1..n}, Q_{1..n}) = ||P_n - Q_n|| + \min \begin{cases} DTW(P_{1..n-1}, Q_{1..m-1}) \\ DTW(P_{1..n-1}, Q_{1..m}) \\ DTW(P_{1..n}, Q_{1..m-1}) \end{cases}$$

όπου  $P_{1..n-1}$  η υποτροχιά του  $P_{1..n}$  που περιλαμβάνει τα σημεία για τις χρονικές στιγμές από 1 έως και  $n-1$ .

Η απόσταση μεγαλύτερης κοινής υπακολουθίας (Longest Common SubSequence, LCSS) χρησιμοποιώντας δυναμικό προγραμματισμό ταιριάζει βέλτιστα τα σημεία των δυο τροχιών με βάση μια παράμετρο ανοχής χρόνου  $\delta$  και μια παράμετρο ανοχής χώρου  $\epsilon$  [VGK02]. Συγκεκριμένα, θεωρεί πως τα σημεία που δεν υπερβαίνουν τις παραμέτρους ανοχής ταιριάζουν και τους αποδίδει τιμή ομοιότητας ίση με 1. Αν δεν ταιριάζουν, τους αποδίδει τιμή ομοιότητας 0.

$$LCSS(P_{1..n}, Q_{1..n}) = \begin{cases} 0 & , \text{ αν } n = 0 \text{ ή } m = 0 \\ 1 + LCSS(P_{1..n-1}, Q_{1..m-1}), & \text{ αν } |n - m| \leq \delta \text{ και } ||P_n - Q_n|| \leq \epsilon \\ \max\{LCSS(P_{1..n-1}, Q_{1..m}), LCSS(P_{1..n}, Q_{1..m-1})\}, & \text{ αλλιώς} \end{cases}$$

Ένα μειονέκτημα του συγκεκριμένου μέτρου απόστασης είναι ότι αγνοεί κενά μεταξύ των υποτροχιών, δηλαδή επιτρέπει να παραληφθούν κάποια σημεία της τροχιάς κατά την ευθυγράμμιση. Κατά συνέπεια, μπορεί να οδηγήσει σε ανακρίβειες κατά την ανάλυση ομοιότητας.



**Σχήμα 2.2:** Παράδειγμα παράλειψης κενών από την απόσταση LCSS

Η απόσταση Edit Distance on Real sequence (EDR) ορίζει την ομοιότητα τροχιών P, Q μήκους n και m αντίστοιχα με βάση τον αριθμό των εισαγωγών, διαγραφών ή αντικαταστάσεων που απαιτούνται ώστε να ταιριάξουν οι δυο τροχιές σύμφωνα με ένα κατώφλι ταιριάσματος  $\varepsilon$  [COO05]. Η απόσταση αυτή χρησιμοποιεί κβαντισμένη τιμή κόστους σε περίπτωση μη ταιριάσματος ή κενών κατά την ευθυγράμμιση και προσπαθεί να ελαχιστοποιήσει το συνολικό κόστος.

$$EDR(P, Q) = \begin{cases} n, & \text{αν } m = 0 \\ m, & \text{αν } n = 0 \\ \min \left\{ \begin{array}{l} EDR(\text{Rest}(P), \text{Rest}(Q)) + \text{subcost}, \\ EDR(\text{Rest}(P), Q) + 1, \\ EDR(P, \text{Rest}(Q)) + 1 \end{array} \right\} & \text{αλλιώς} \end{cases}$$

$$\text{όπου subcost} = \begin{cases} 0, & \text{αν } |p_{i,x} - q_{j,x}| \leq \varepsilon \text{ και } |p_{i,y} - q_{j,y}| \leq \varepsilon \\ 1 & \text{αλλιώς} \end{cases}$$

Τέλος, η απόσταση Edit Distance with Real Penalty (ERP) αποτελεί τη μοναδική απόσταση που βασίζεται σε ευθυγράμμιση και ικανοποιεί την τριγωνική ανισότητα, επιτρέποντας έτσι να γίνει εύκολα η δεικτοδότηση της. Η απόσταση αυτή ορίζεται όμοια με την απόσταση EDR, αλλά χρησιμοποιεί την απόσταση Manhattan για υποτροχιές που δεν περιέχουν κενά, καθώς και μια σταθερή τιμή g για τον υπολογισμό της απόστασης στην περίπτωση που υπάρχουν κενά [CN04].

$$ERP(P, Q) = \begin{cases} \sum_1^n |q_i - g|, & \text{αν } m = 0 \\ \sum_1^m |p_i - g|, & \text{αν } n = 0 \\ \min \left\{ \begin{array}{l} ERP(\text{Rest}(P), \text{Rest}(Q)) + \text{dist}_{erp}(p_1, q_1), \\ ERP(\text{Rest}(P), Q) + \text{dist}_{erp}(p_1, \text{gap}), \\ EDR(P, \text{Rest}(Q)) + \text{dist}_{erp}(q_1, \text{gap}) \end{array} \right\} & \text{αλλιώς} \end{cases}$$

$$\text{όπου dist}_{erp}(p_i, q_i) = \begin{cases} |p_i - q_i| \text{ αν τα } p_i, q_i \text{ δεν περιέχουν κενά} \\ |p_i - g| \text{ αν το } q_i \text{ περιέχει κενό} \\ |q_i - g| \text{ αν το } p_i \text{ περιέχει κενό} \end{cases}$$

Συνοψίζοντας, τα παραπάνω μέτρα απόστασης που επιτρέπουν ευθυγράμμιση υπολογίζουν τη χωρική ομοιότητα των τροχιών αγνοώντας την παράμετρο του χρόνου. Επίσης, βασιζόμενα ουσιαστικά στην ευκλείδεια απόσταση, συγκρίνουν συνολικά το σχήμα των τροχιών, χωρίς να λαμβάνουν υπόψη την τοπική ομοιότητα των υποτροχιών και εγγενή χαρακτηριστικά των τροχιών.

## Απόσταση που βασίζεται σε ελάχιστα περιβάλλοντα ορθογώνια

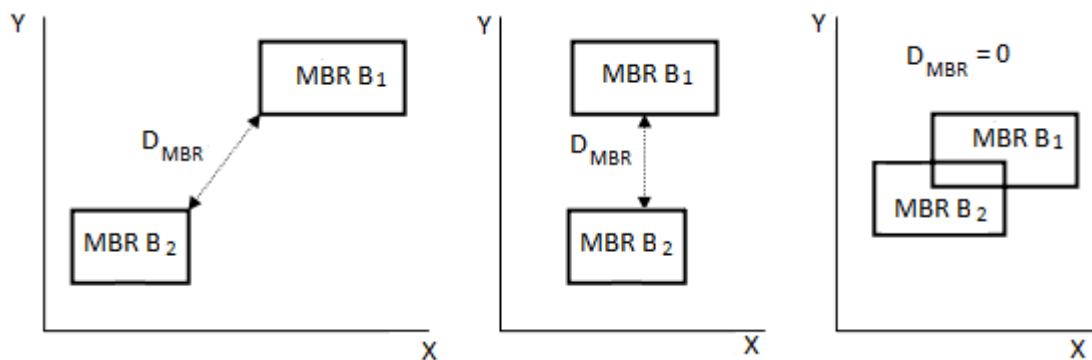
Για την ανάλυση ομοιότητας τροχιών χρησιμοποιείται συχνά η απόσταση μεταξύ ελάχιστων περιβαλλόντων ορθογώνιων (Minimum Bounding Rectangles, MBRs) [LCK+00, ZZ11]. Τα ελάχιστα περιβάλλοντα ορθογώνια χρησιμεύουν στην προσεγγιστική αναπαράσταση τμημάτων μιας τροχιάς, στην αποδοτική δεικτοδότηση τους με κάποια παραλλαγή των R-δέντρων, καθώς και στον ταχύ υπολογισμό των αποστάσεων μεταξύ των τροχιών.

Η απόσταση μεταξύ δύο MBRs  $B_1$ ,  $B_2$  ορίζεται ως η ελάχιστη ευκλείδεια απόσταση μεταξύ οποιουδήποτε ζεύγους σημείων των  $B_1$  και  $B_2$ :

$$D_{\min}(B_1, B_2) = \sqrt{(\Delta(B_1 \cdot [x_l, x_u], B_2 \cdot [x_l, x_u]))^2 + (\Delta(B_1 \cdot [y_l, y_u], B_2 \cdot [y_l, y_u]))^2},$$

όπου η απόσταση μεταξύ δύο διαστημάτων ορίζεται ως:

$$\Delta([l_1, u_1], [l_2, u_2]) = \begin{cases} 0 & , \quad \text{αν } [l_1, u_1] \cap [l_2, u_2] \neq \emptyset \\ l_2 - u_1 & , \quad \text{αν } u_1 < l_2 \\ l_1 - u_2 & , \quad \text{αν } u_2 < l_1 \end{cases}$$



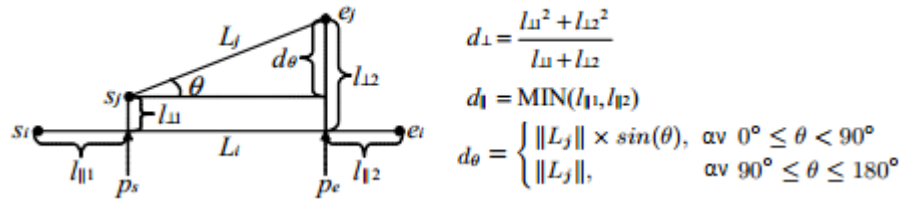
Σχήμα 2.3: Παραδείγματα απόστασης μεταξύ MBRs

Η απόσταση που βασίζεται σε ελάχιστα περιβάλλοντα ορθογώνια υπολογίζει προσεγγιστικά την απόσταση μεταξύ δυο τροχιών κι ενδείκνυται ως προεπεξεργασία για κλάδεμα τροχιών που διαφέρουν σημαντικά, αλλά όχι για ανάλυση ομοιότητας τροχιών με ακρίβεια.

## Hausdorff απόσταση τροχιάς

Η Hausdorff απόσταση τροχιάς [LHW07, ZZ11] ορίζεται ως άθροισμα τριών συνιστωσών:

- η κάθετη απόσταση ( $d_{\perp}$ ) εκφράζει το διαχωρισμό (separation) των τροχιών
- η παράλληλη απόσταση ( $d_{\parallel}$ ) εκφράζει τη διαφορά μήκους των τροχιών
- η γωνιακή απόσταση ( $d_{\theta}$ ) εκφράζει τη διαφορά κατεύθυνσης των τροχιών



Σχήμα 2.4: Οι τρεις συνιστώσες της Hausdorff απόστασης τροχιάς

$D_{\text{Hausdorff}} = w_{\perp} \cdot d_{\perp} + w_{\parallel} \cdot d_{\parallel} + w_{\theta} \cdot d_{\theta}$ , όπου  $w_{\perp}$ ,  $w_{\parallel}$ , and  $w_{\theta}$  είναι τα βάρη των αντίστοιχων συνιστωσών. Διαφορετικές εφαρμογές μπορεί να απαιτούν διαφορετικά βάρη, αλλά η συνήθης τους τιμή είναι 1.

Η Hausdorff απόσταση τροχιάς αποτελεί καταλληλότερη συνάρτηση απόστασης για ανάλυση ομοιότητας, καθώς λαμβάνει υπόψη εγγενή χαρακτηριστικά της τροχιάς καθώς και την τοπική ομοιότητα των υποτροχιών.

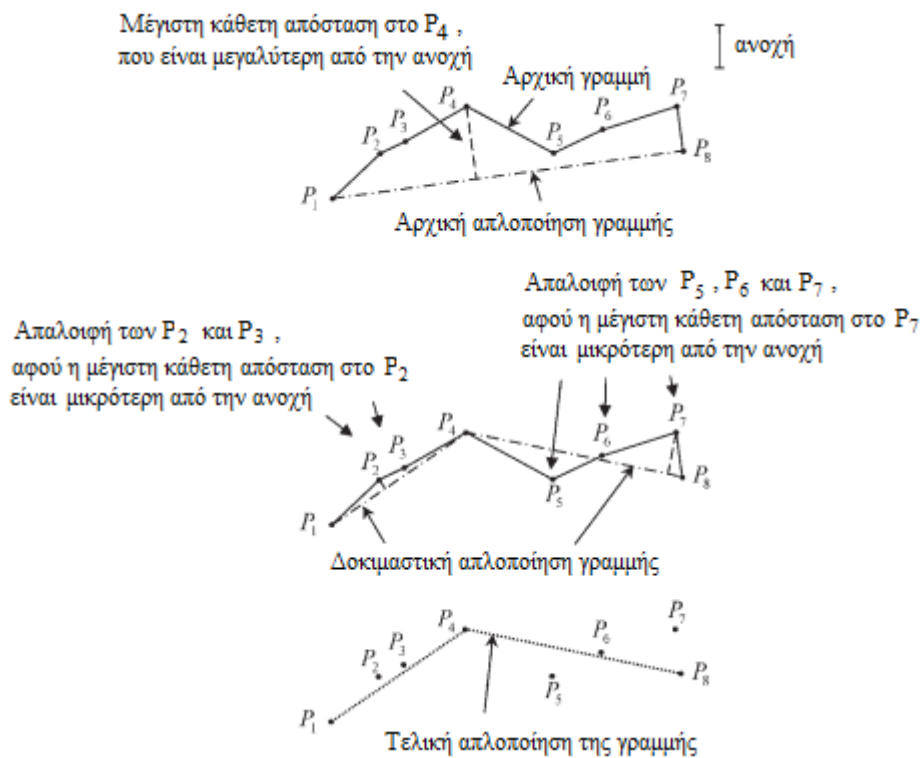
## 2.2 Απλοποίηση τροχιών

Η απλοποίηση τροχιών έχει σκοπό την αφαίρεση πλεονάζουσας πληροφορίας από τα γεωγραφικά πληροφοριακά συστήματα (geographical information systems – GIS). Η απλοποίηση μπορεί να γίνει με τοπικές [L69, RW74, O81, ZS97] ή καθολικές μεθόδους [DP73, VW93]. Οι τοπικές μέθοδοι επεξεργάζονται με ακολουθιακό τρόπο τμήματα της τροχιάς και μπορούν επίσης να χρησιμοποιηθούν για online επεξεργασία, αφού δεν απαιτούν γνώση ολόκληρης της τροχιάς. Αντίθετα, οι καθολικές μέθοδοι εξετάζουν ολόκληρη την τροχιά για την εξομάλυνση της και παρέχουν συνήθως πιο ακριβή και αξιόπιστα αποτελέσματα.

Line simplification Algorithm	Visual difference	Displacement in the 'certain' case	Displacement in the 'uncertain' case	Shape distortion in the 'certain' case	Shape distortion in the 'uncertain' case	Reciprocal of the computation time
Douglas–Peucker [DP73]	***	***	***	***	***	*
Zhao–Saalfeld [ZS97]	***	**	**	**	*	***
Reumann–Witkam [RW74]	***	**	**	**	**	*
Visvalingham–Whyatt [VW93]	***	**	**	**	*	*
Opheim [O81]	***	*	*	*	*	*
Lang [L69]	***	**	**	**	*	**

Σχήμα 2.5: Σύγκριση των αλγορίθμων απλοποίησης τροχιάς [SC06]

Πράγματι, η αξιολόγηση επίδοσης των αλγορίθμων απλοποίησης τροχιών [SC06] δείχνει πως τα ακριβέστερα αποτελέσματα επιτυγχάνονται με τον αλγόριθμο Douglas–Peucker [DP73]. Αυτός ο αλγόριθμος διαιρεί αναδρομικά την πολυγραμμή (polyline) της τροχιάς. Αρχικά βρίσκει το σημείο με τη μέγιστη απόσταση από το ευθύγραμμο τμήμα  $L$  που ενώνει τα άκρα της τροχιάς. Αν αυτό το σημείο απέχει περισσότερο από την απόσταση ανοχής  $\epsilon$ , τότε ο αλγόριθμος εκτελείται πάλι για το τμήμα της τροχιάς μεταξύ του αρχικού σημείου και αυτού του σημείου και μεταξύ αυτού του σημείου και του τελικού σημείου. Αλλιώς, επιστρέφει το ευθύγραμμο τμήμα  $L$ .



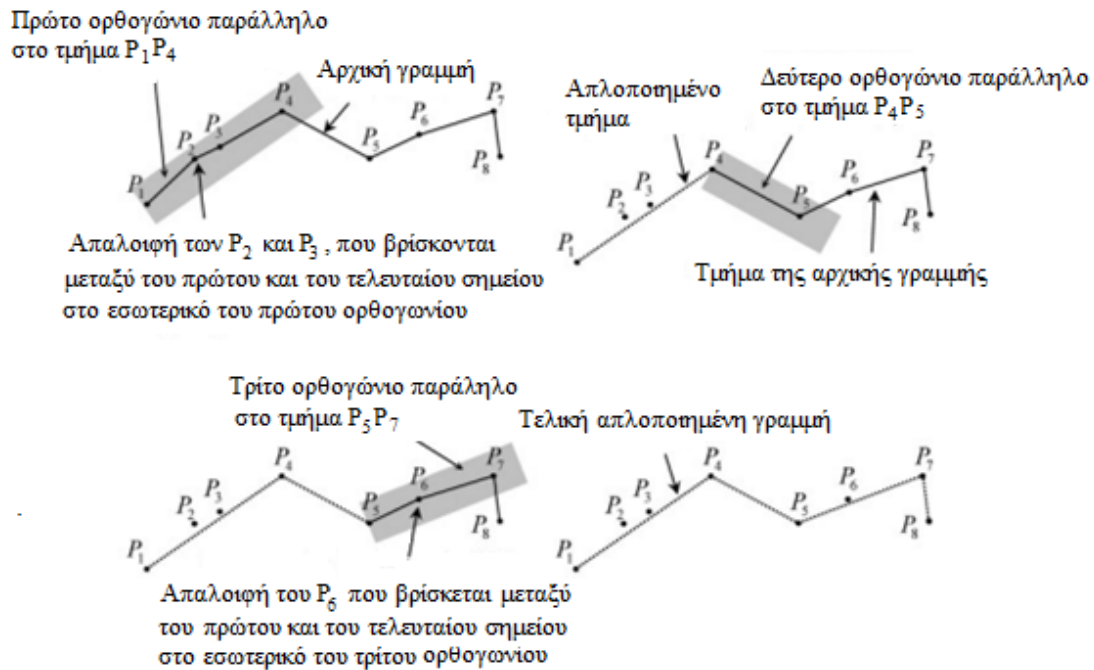
**Σχήμα 2.6:** Παράδειγμα απλοποίησης γραμμής με τον αλγόριθμο Douglas–Peucker

Ωστόσο, ο αλγόριθμος παρουσιάζει μεγάλο υπολογιστικό κόστος. Η πολυπλοκότητα του είναι  $\Theta(n \cdot \log n)$  και στη χειρότερη περίπτωση μπορεί να γίνει  $O(n^2)$ .

Αν απαιτείται μια γρήγορη μέθοδος απλοποίησης γραμμής, ο αλγόριθμος Zhao – Saalfeld [ZS97] έχει μεγαλύτερη χρησιμότητα, αφού μπορεί να απλοποιήσει μια τροχιά σε γραμμικό χρόνο με μεγαλύτερη ακρίβεια από άλλους αλγορίθμους απλοποίησης.

Ο συγκεκριμένος αλγόριθμος προσπαθεί να χωρέσει τα πρώτα  $m$  σημεία μιας πολυγραμμής σε ένα ορθογώνιο πλάτους  $2\epsilon$  και άπειρου μήκους. Αν ο αλγόριθμος χωρέσει τα πρώτα  $k$  σημεία, πρέπει να περιστρέψει το ορθογώνιο για να ελέγξει αν χωράει και το  $(k+1)$ -οστό

σημείο. Διατηρώντας όμως το πρώτο σημείο στο κέντρο του ορθογωνίου και τα επόμενα σημεία στο εσωτερικό του, το ορθογώνιο περιορίζεται στην περιστροφή του.



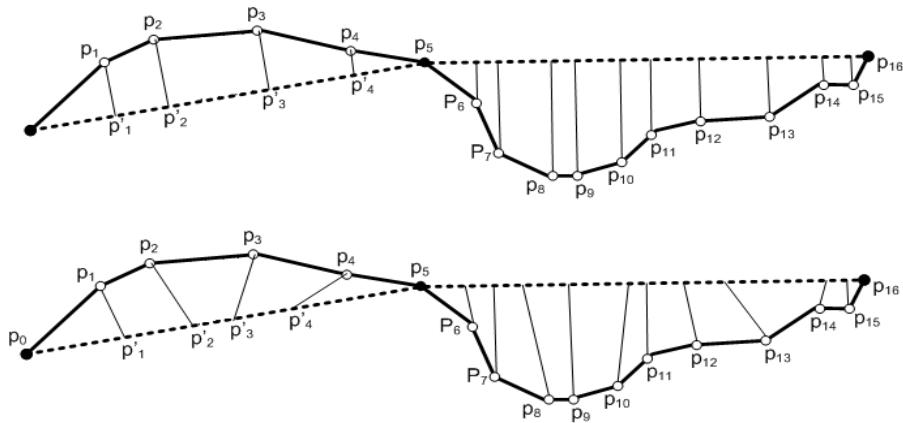
Σχήμα 2.7: Παράδειγμα απλοποίησης γραμμής με τον αλγόριθμο Zhao–Saalfeld

### Χωρο-χρονική επέκταση

Οι παραπάνω αλγόριθμοι δε λαμβάνουν υπόψη το χρόνο, αφού η απόσταση ανοχής βασίζεται στην ευκλείδεια απόσταση, δηλαδή στην προβολή των σημείων της αρχικής τροχιάς στην απλοποιημένη τροχιά. Για να συμπεριληφθεί ο χρονικός παράγοντας, μπορούμε να χρησιμοποιήσουμε τη συγχρονισμένη ευκλείδεια απόσταση (time synchronized euclidean distance), η οποία υποθέτει ότι τα αντικείμενα κινούνται με σταθερή ταχύτητα στα απλοποιημένα τμήματα της τροχιάς κι έτσι συγχρονίζει τα αρχικά σημεία με τα αντίστοιχα σημεία στην απλοποιημένη τροχιά με βάση το χρόνο.

Όμως, η χρήση της συγχρονισμένης ευκλείδειας απόστασης στους αλγόριθμους απλοποίησης απαιτεί  $O(n^2)$  υπολογιστική πολυπλοκότητα [MB04]. Για να διατηρήσουμε το γραμμικό χρόνο εκτέλεσης, μπορούμε εναλλακτικά να λάβουμε υπόψη την κατεύθυνση και την ταχύτητα [PPT06]. Αφού ο περιορισμός στην κατεύθυνση υπάρχει στον αλγόριθμο του Zhao – Saalfeld, αρκεί να προστεθεί περιορισμός στην ταχύτητα.





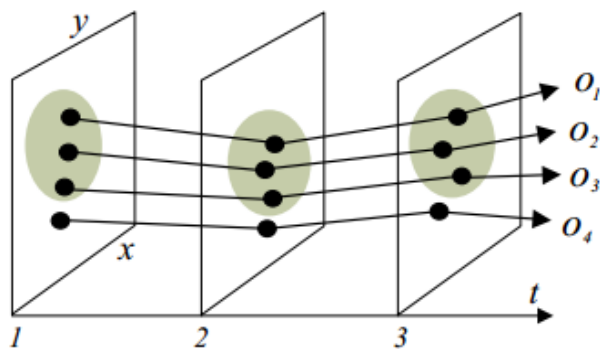
**Σχήμα 2.8:** Παράδειγμα απόστασης σημείου από ευθεία με βάση την ευκλείδεια απόσταση και τη συγχρονισμένη ευκλείδεια απόσταση

## 2.3 Εύρεση ομάδων κινούμενων αντικειμένων

Στη βιβλιογραφία έχουν προταθεί διαφορετικοί ορισμοί για τις ομάδες κινούμενων αντικειμένων, όπως flocks, convoys, moving clusters, group patterns και swarms.

Flock ορίζεται μια ομάδα αντικειμένων που κινούνται μαζί για τουλάχιστον  $k$  διαδοχικές χρονικές στιγμές, παραμένοντας σε ένα δίσκο ακτίνας  $r$  [GK08, VBT09].

Όπως απεικονίζεται στο σχήμα 2.9, ένα πρόβλημα με τον ορισμό αυτό είναι η δύσκολη επιλογή κατάλληλης ακτίνας δίσκου που να περιλαμβάνει όλα τα αντικείμενα τα οποία κινούνται μαζί. Επίσης, είναι προτιμότερο να μην ορίζεται εκ των προτέρων σταθερό σχήμα κίνησης της ομάδας, αφού τα αντικείμενα μπορεί να κινούνται κοντά άλλα άτακτα.

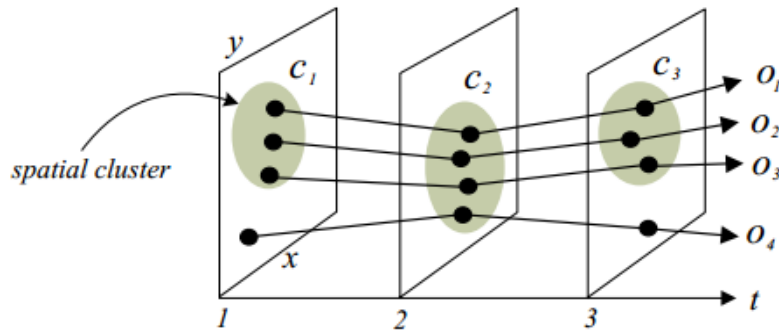


**Σχήμα 2.9:** Παράδειγμα ενός flock κινούμενων αντικειμένων

Moving cluster [KNB05] είναι ένα σύνολο χωρικών ομάδων (spatial clusters) που εμφανίζονται σε διαδοχικές χρονικές στιγμές και το ποσοστό κοινών αντικειμένων σε οποιοσδήποτε δύο διαδοχικές τέτοιες ομάδες δεν είναι μικρότερο από ένα δοθέν κατώφλι  $\theta$ :

$$\frac{|c_t \cap c_{t+1}|}{|c_t \cup c_{t+1}|} \geq \theta, \text{ όπου } c_t \text{ μια ομάδα τη χρονική στιγμή } t.$$

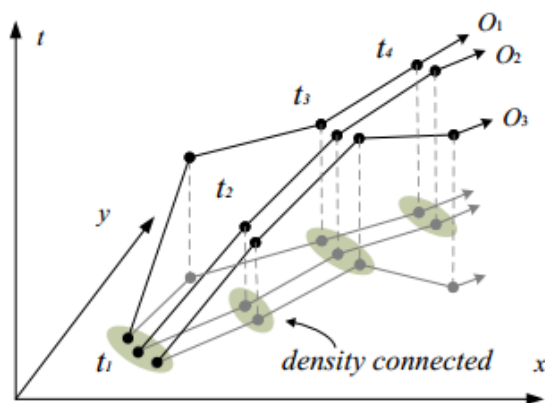
Αυτός ο ορισμός επιτρέπει να μεταβάλλονται τα μέλη της ομάδας στο χρόνο, αρκεί κάποια από αυτά να παραμένουν ίδια σε διαδοχικές χρονικές στιγμές.



Σχήμα 2.10: Παράδειγμα ενός moving cluster (κατώφλι  $\theta = 0,75$ )

Convoy είναι ένα σύνολο αντικειμένων που είναι πυκνά συνδεδεμένα σύμφωνα με ένα κατώφλι απόστασης  $\epsilon$  για τουλάχιστον  $k$  διαδοχικές χρονικές στιγμές [JSZ08, JYZ08]. Δύο παραλλαγές του ορισμού είναι τα δυναμικά convoys (dynamic convoys), που επιτρέπουν στα μέλη να απομακρύνονται προσωρινά από την ομάδα, καθώς και τα εξελισσόμενα convoys (evolving convoys), τα οποία μπορούν να συρρικνώνονται και να επεκτείνονται ως προς τον αριθμό των μελών τους [AT10].

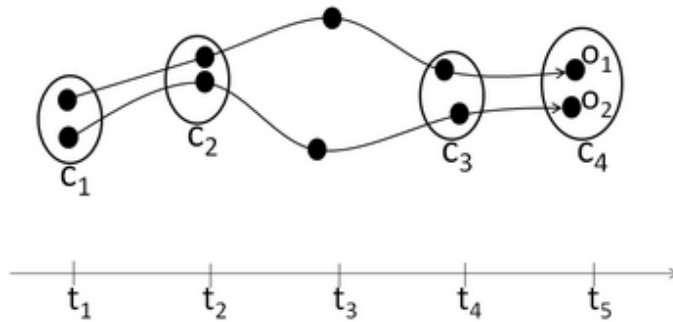
Ο ορισμός του convoy αποτελεί επέκταση του ορισμού του flock, αφού επιτρέπει οι χωρικές ομάδες να έχουν αυθαίρετο σχήμα και μέγεθος.



Σχήμα 2.11: Παράδειγμα ενός convoy

Group pattern είναι ένα σύνολο αντικειμένων που κινούνται μαζί, παραμένοντας σε ένα δίσκο ακτίνας  $r$ , για κάποιες πιθανόν μη διαδοχικές χρονικές στιγμές [WLH06].

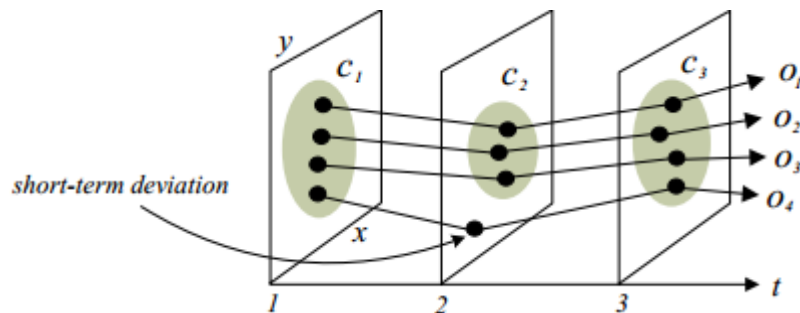
Ο ορισμός αυτός περιορίζει το σχήμα και το μέγεθος των χωρικών ομάδων, αλλά επιτρέπει στα αντικείμενα να απομακρύνονται προσωρινά από την ομάδα.



Σχήμα 2.12: Παράδειγμα ενός group pattern

Swarm είναι ένα σύνολο αντικειμένων που είναι πυκνά συνδεδεμένα σύμφωνα με ένα κατάφλι απόστασης  $e$  για τουλάχιστον  $k$  πιθανόν μη διαδοχικές χρονικές στιγμές [LDH+10].

Αυτός ο ορισμός είναι ο πιο γενικός, αφού επιτρέπει οι χωρικές ομάδες να έχουν αυθαίρετο σχήμα και μέγεθος, καθώς και τα μέλη της ομάδας να απομακρύνονται προσωρινά.



Σχήμα 2.13: Παράδειγμα ενός swarm

Συνοψίζοντας, οι ομάδες κινούμενων αντικειμένων περιλαμβάνουν αντικείμενα που κινούνται σε κοντινή απόσταση μεταξύ τους σε σχέση με ένα κατάφλι απόστασης. Αντιθέτως, στην παρούσα εργασία αναζητούμε αντικείμενα που κινούνται σε κοντινή απόσταση από ένα δοθέν, χωρίς να εισάγουμε περιορισμούς απόστασης.

## 2.4 Εύρεση k-NN κινούμενων αντικείμενων

Τα ερωτήματα εύρεσης κοντινότερων γειτόνων αποτελούν, μαζί με τα ερωτήματα περιοχής (range queries), τα πιο θεμελιώδη ερωτήματα σε χωρικές βάσεις δεδομένων. Τα ερωτήματα διακρίνονται σε τέσσερις κατηγορίες [GBX10]:

- στατικό ερώτημα σε στατικά αντικείμενα (το κλασικό πρόβλημα κοντινότερων γειτόνων)
- κινούμενο ερώτημα σε στατικά αντικείμενα (π.χ τα κοντινότερα ξενοδοχεία για έναν ταξιδιώτη)
- στατικό ερώτημα σε κινούμενα ερωτήματα (π.χ τα κοντινότερα ασθενοφόρα σε ένα χώρο ατυχήματος)
- κινούμενο ερώτημα σε κινούμενα ερωτήματα (π.χ τα κοντινότερα οχήματα στο όχημα της καγκελαρίου Μέρκελ κατά το ταξίδι της στην Αθήνα)

Το τελευταίο είδος ερωτημάτων είναι το πιο δύσκολο και το πιο γενικό. Συμπεριλαμβάνει τις υπόλοιπες κατηγορίες ερωτημάτων, αφού ένα στάσιμο αντικείμενο μπορεί να αναπαρασταθεί ως κινούμενο που μένει στο ίδιο σημείο.

Επίσης, η εύρεση κοντινότερων γειτόνων σε κάποιο χρονικό διάστημα μπορεί να μελετηθεί με συνεχή ή μη συνεχή τρόπο. Η συνεχής προσέγγιση χρησιμοποιεί κάποια συναθροιστική συνάρτηση ώστε να επιστρέφει τους κοντινότερους γείτονες σε όλο το διάστημα, ενώ η μη συνεχής προσέγγιση επιστρέφει τα αντικείμενα που ήταν κοντινότεροι γείτονες σε οποιαδήποτε χρονική στιγμή στο δοθέν διάστημα [FGP+07].

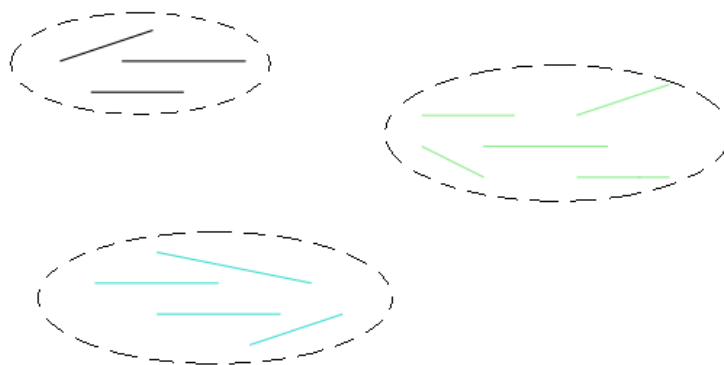
Τα ερωτήματα που μας ενδιαφέρουν στην παρούσα εργασία περιλαμβάνουν κινούμενα ερωτήματα σε κινούμενα αντικείμενα και θα πραγματοποιηθούν με συναθροιστικό τρόπο. Τα συναθροιστικά ερωτήματα κοντινότερων γειτόνων επιστρέφουν τα αντικείμενα που ελαχιστοποιούν μια συναθροιστική συνάρτηση απόστασης (όπως min, max ή sum) σε σχέση με ένα δοθέν σύνολο σημείων [PTM+05]. Θεωρώντας τα χωρο-χρονικά χαρακτηριστικά των τροχιών, ορίζουμε το δοθέν σύνολο σημείων ως ένα δοθέν σύνολο τμημάτων τροχιάς σε διαδοχικά χρονικά διαστήματα και τη συναθροιστική απόσταση ως το άθροισμα της ομοιότητας των αντίστοιχων τμημάτων τροχιάς κατά τη διάρκεια αυτών των χρονικών διαστημάτων.

## 2.5 k-NN σε προβλήματα εξόρυξης γνώσης από τροχιές

Πέρα από τα αυτοτελή ερωτήματα k-NN σε βάσεις δεδομένων, η εύρεση κοντινότερων γειτόνων συμβάλλει στην επίλυση διαφόρων προβλημάτων εξόρυξης γνώσης. Στη συνέχεια αναφέρουμε συνοπτικά τα σημαντικότερα από αυτά.

### Ομαδοποίηση (Clustering)

Η ομαδοποίηση τροχιών έχει ως στόχο την αναγνώριση ομάδων στις οποίες οι τροχιές ακολουθούν όμοια πρότυπα κίνησης. Η πιο συνηθισμένη προσέγγιση καθορίζει τις ομάδες με βάση την πυκνότητα των τροχιών σε κάποιο χώρο (density-based clustering) [LHW07].

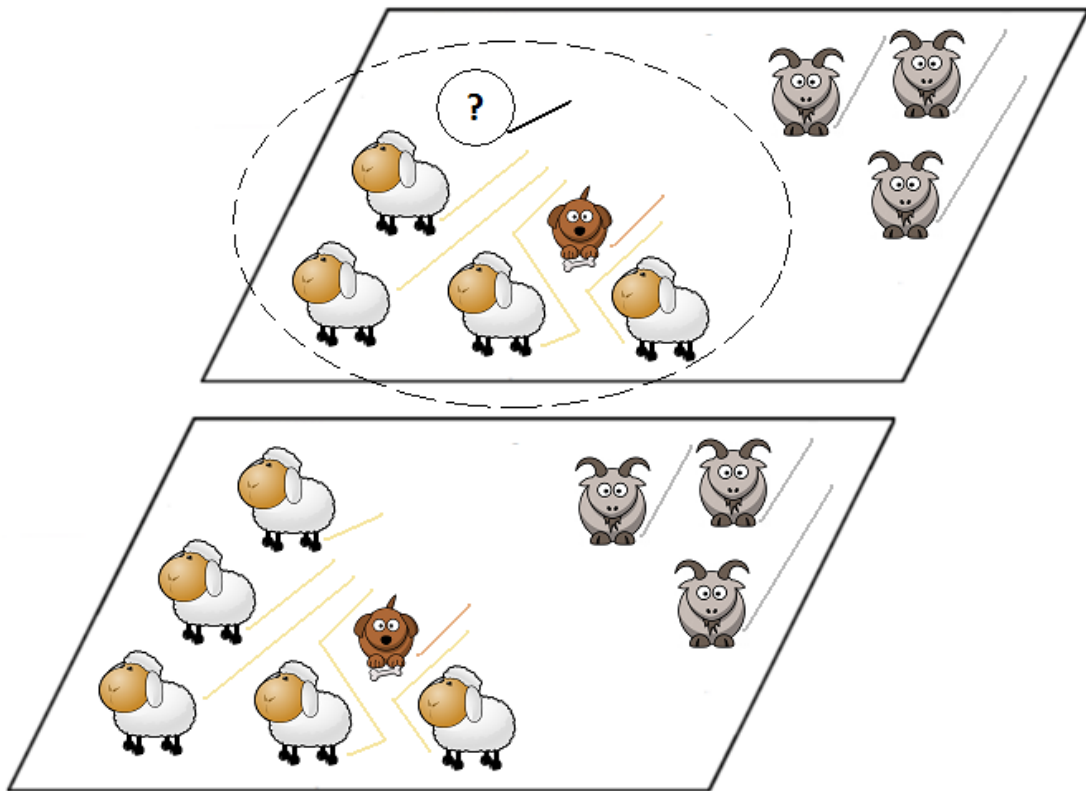


Σχήμα 2.9: Παράδειγμα ομαδοποίησης τροχιών με βάση την πυκνότητα

Ωστόσο, αυτή η μέθοδος υποθέτει ότι οι τροχιές σε κάθε ομάδα έχουν αρκετά ομοιόμορφη πυκνότητα, το οποίο δεν ισχύει πάντα σε πραγματικά σύνολα δεδομένων. Γι' αυτό, μια εναλλακτική προσέγγιση καθορίζει τις ομάδες συγκρίνοντας το ποσοστό ίδιων κ κοντινότερων γειτόνων δύο τροχιών κι έπειτα συγχωνεύει ομάδες που απέχουν μεταξύ τους λιγότερο από ένα δοθέν κατώφλι απόστασης [RH10].

### Κατηγοριοποίηση (Classification)

Η κατηγοριοποίηση τροχιών έχει ως στόχο να αναθέσει μια τροχιά σε ένα προκαθορισμένο σύνολο κατηγοριών. Μια απλή προσέγγιση είναι ανάθεση της τροχιάς στην κατηγορία στην οποία ανήκουν οι περισσότεροι από τους κ κοντινότερους γείτονες [SVS10]. Ο αλγόριθμος kNN βασίζεται σε οκνηρή μάθηση (lazy learning), αφού δεν απαιτεί προηγούμενη εκπαίδευση αλλά χρησιμοποιεί απευθείας το σύνολο δεδομένων εκπαίδευσης (training set), δηλαδή τις υπόλοιπες τροχιές, όταν ζητηθεί κατηγοριοποίηση ενός αντικειμένου.



Σχήμα 2.10: Παράδειγμα κατηγοριοποίησης τροχιών με kNN

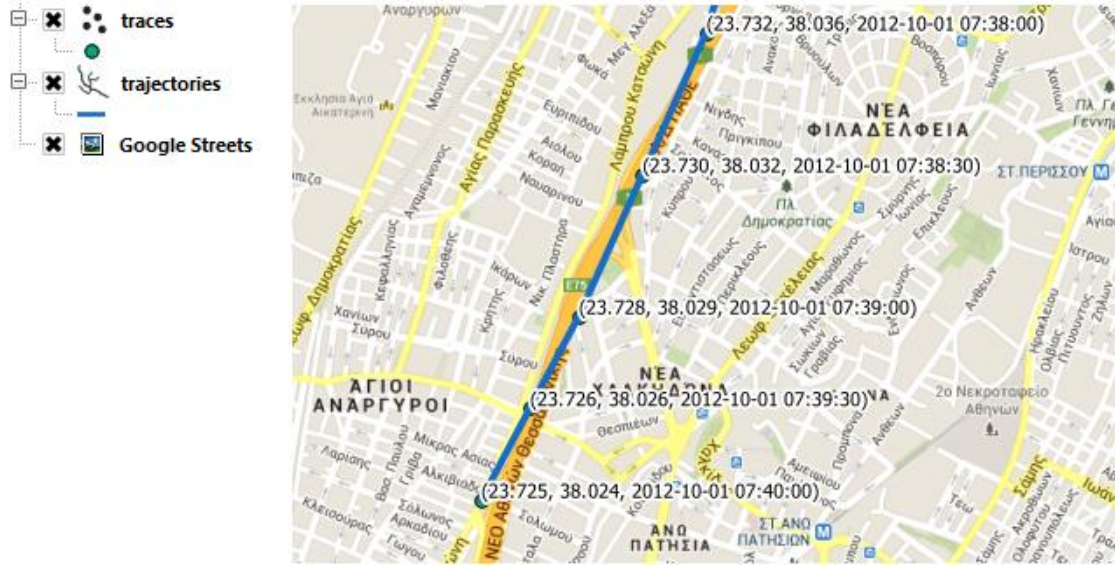
# 3

## Ορισμός προβλήματος

Στο κεφάλαιο αυτό περιγράφεται αναλυτικά *το πρόβλημα που πραγματεύεται* η διπλωματική και εξηγούνται συνοπτικά έννοιες που θα συμβάλλουν στην καλύτερη κατανόηση του προβλήματος.

### 3.1 Προαπαιτούμενα

**Τροχιά** ενός κινούμενου αντικειμένου είναι το σύνολο των διαδοχικών θέσεων του στο χώρο ως συνάρτηση του χρόνου. Λόγω εγγενών περιορισμών στην απόκτηση και στην αποθήκευση δεδομένων, είναι πολύ δύσκολη και δαπανηρή η ακριβής καταγραφή ολόκληρης της τροχιάς. Κατά συνέπεια, στις χωρο-χρονικές βάσεις δεδομένων η τροχιά αναπαρίσταται ως ένα σύνολο διακριτών δειγμάτων των θέσεων ενός αντικειμένου της μορφής  $(x,y,t)$ , δηλαδή μια ακολουθία σημείων  $(x,y)$  που το καθένα χαρακτηρίζεται από ένα χρονόσημο (timestamp).



Σχήμα 2.4: Παράδειγμα αναπαράστασης τροχιάς κινούμενου αντικειμένου

Η χωρο-χρονική ομοιότητα τροχιών καθορίζεται από την απόσταση των δύο τροχιών στις αντίστοιχες χρονικές στιγμές. Συγκεκριμένα, θεωρώντας τις τροχιές ως σύνολα διαδοχικών σημείων στο χρόνο  $P_{t_1..t_n}$ ,  $Q_{t_1..t_n}$ , η ομοιότητά τους ορίζεται ως:

$$\text{sim}_{t_i,t_{i+1}}(P, Q) = \frac{1}{1 + \text{dist}(P_{t_i,t_{i+1}}, Q_{t_i,t_{i+1}})},$$

όπου  $t_i \in [t_1, \dots, t_{n-1}]$

και  $\text{dist}(P_{t_i,t_{i+1}}, Q_{t_i,t_{i+1}})$  η Hausdorff απόσταση τροχιάς.

Η ομοιότητα  $\text{sim}$  λαμβάνει τιμές από 0 έως 1. Αυτός ο ορισμός ομοιότητας λαμβάνει υπόψη την περίπτωση που δεν ορίζεται η τροχιά για κάποιο χρονικό διάστημα, οπότε η απόσταση των τροχιών θεωρείται άπειρη και η ομοιότητα μηδενική.

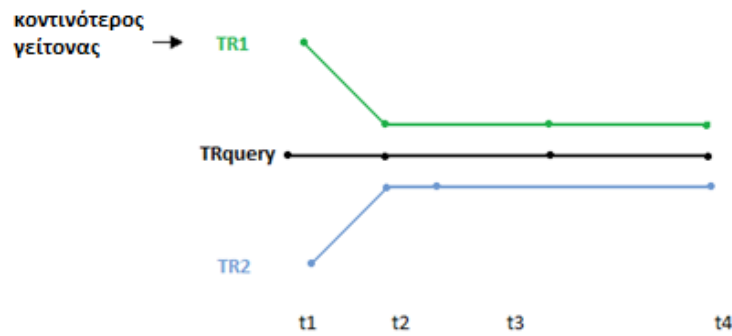
### 3.2 Το πρόβλημα εύρεσης κ κοντινότερων γειτόνων σε ιστορικές βάσεις δεδομένων

Δοθέντος ενός συνόλου τροχιών  $\mathcal{I}$ , μιας τροχιάς  $TR_{\text{query}}$  κι ενός χρονικού παραθύρου  $T = \{t_1, t_2, \dots, t_n\}$ , θέλουμε να βρούμε ένα υποσύνολο  $\mathcal{I}'$  του  $\mathcal{I}$  με  $k$  τροχιές (δηλαδή  $\mathcal{I}' \subseteq \mathcal{I}$ ,  $|\mathcal{I}'| = k$ ), έτσι ώστε  $\forall TR_i \in \mathcal{I}'$ ,  $\forall TR_j \in \mathcal{I} - \mathcal{I}'$ :

$$\frac{\sum_{t_i=t_1}^{t_{n-1}} \text{sim}_{t_i,t_{i+1}}(TR_{\text{query}}, TR_i)}{|\mathcal{I}'|-1} \geq \frac{\sum_{t_i=t_1}^{t_{n-1}} \text{sim}_{t_i,t_{i+1}}(TR_{\text{query}}, TR_j)}{|\mathcal{I}|-1}.$$



Για την εύρεση των κ κοντινότερων γειτόνων συγκρίνουμε το μέσο όρο της ομοιότητας των υποτροχιών κατά τα αντίστοιχα χρονικά διαστήματα, όπως φαίνεται στο σχήμα 2.4.



$$\text{sim}_{\text{TR1}} = \frac{\text{sim}_{t1,t2}(\text{TRquery}, \text{TR1}) + \text{sim}_{t2,t3}(\text{TRquery}, \text{TR1}) + \text{sim}_{t3,t4}(\text{TRquery}, \text{TR1})}{3} = 0,85$$

$$\text{sim}_{\text{TR2}} = \frac{\text{sim}_{t1,t2}(\text{TRquery}, \text{TR2}) + \text{sim}_{t2,t3}(\text{TRquery}, \text{TR2}) + \text{sim}_{t3,t4}(\text{TRquery}, \text{TR2})}{3} = 0,82$$

**Σχήμα 2.4:** Παράδειγμα σύγκρισης ομοιότητας τροχιών για την εύρεση του κοντινότερου γείτονα



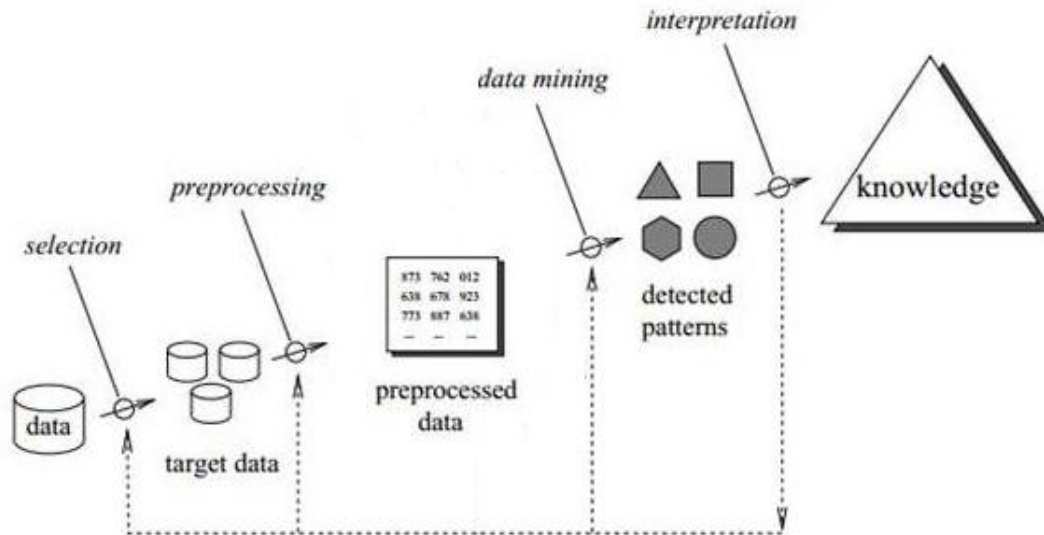
# 4

## Μέθοδοι επίλυσης του προβλήματος

Στο κεφάλαιο αυτό περιγράφονται οι αλγόριθμοι για την επίλυση του προβλήματος. Συγκεκριμένα, αναλύεται η απαιτούμενη προεπεξεργασία των δεδομένων και κατόπιν περιγράφεται ο βασικός αλγόριθμος επίλυσης του προβλήματος, μια παραλλαγή του με εισαγωγή ουράς προτεραιότητας και τέλος μια προσεγγιστική μέθοδος επίλυσης.

### 4.1 Προεπεξεργασία δεδομένων

Εξαιτίας τεχνολογικών αδυναμιών σε συστήματα εντοπισμού θέσης, η καταγραφή των τροχιών μπορεί να είναι ανομοιογενής. Για παράδειγμα, μπορεί σε συγκεκριμένες χρονικές στιγμές οι θέσεις των κινούμενων αντικειμένων να είναι άγνωστες σε κάποιες τροχιές αλλά γνωστές σε άλλες. Γι' αυτό το λόγο, η προεπεξεργασία των δεδομένων (data preprocessing) είναι απαραίτητη ώστε τα δεδομένα να αποκτήσουν ομοιογενή μορφή και να μπορούν πλέον να χρησιμοποιηθούν για εξόρυξη γνώσης.



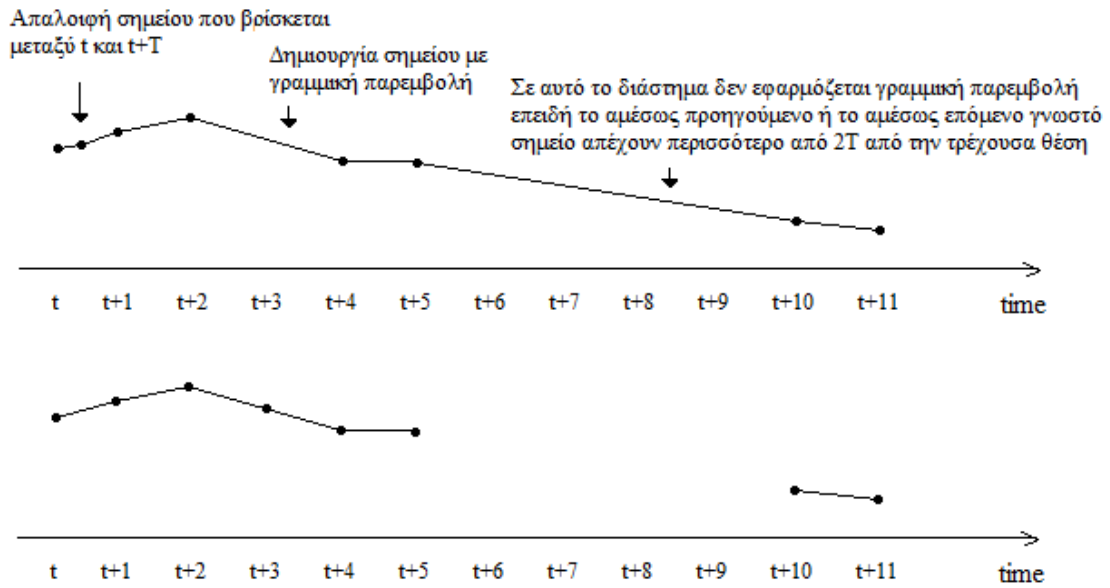
**Σχήμα 4.1:** Τα στάδια της διαδικασίας εξόρυξης δεδομένων

Για να γίνει χωρο-χρονική ανάλυση των τροχιών, η προεπεξεργασία περιλαμβάνει δειγματοληψία στο σύνολο δεδομένων (dataset) που χρησιμοποιείται και παρεμβολή για τα σημεία που λείπουν. Η πιο συχνά χρησιμοποιούμενη μέθοδος παρεμβολής είναι η γραμμική. Σύμφωνα με αυτή, θεωρούμε ότι η κίνηση του αντικειμένου ακολουθεί το γραμμικό μοντέλο και η θέση του  $(x_{t1}, y_{t1})$  τη χρονική στιγμή  $t_1$  μπορεί να βρεθεί από την αμέσως προηγούμενη και την αμέσως επόμενη θέση του ως εξής:

$$(x_{t1}, y_{t1}) = (x_{t0} + \frac{t1-t0}{t2-t0}(x_{t2} - x_{t0}), y_{t0} + \frac{t1-t0}{t2-t0}(y_{t2} - y_{t0}))$$

Ωστόσο, σε περίπτωση που η χρονική στιγμή καταγραφής της προηγούμενης ή της επόμενης θέσης απέχει περισσότερο από 2 φορές την περίοδο δειγματοληψίας, θεωρούμε πως η μέθοδος της παρεμβολής δε δίνει αξιόπιστα αποτελέσματα οπότε η τροχιά του αντικειμένου διακόπτεται για αυτές τις θέσεις που λείπουν.

περίοδος δειγματοληψίας  $T = 1$



Σχήμα 4.1: Παράδειγμα δειγματοληψίας και γραμμικής παρεμβολής

## 4.2 Εύρεση των $k$ κοντινότερων γειτόνων

### 4.2.1 Βασικός αλγόριθμος

Δοθέντος ενός συνόλου τροχιών  $\mathcal{D}$ , μιας τροχιάς  $TR_{query}$ , ενός χρονικού παραθύρου  $T$  κι ενός αριθμού  $k$ , ο βασικός αλγόριθμος επιστρέφει τις  $k$  κοντινότερες τροχιές στην τροχιά  $TR_{query}$ . Αρχικά, ο αλγόριθμος εξετάζει επαναληπτικά όλες τις τροχιές στο  $\mathcal{D}$  και για καθεμία από αυτές, προσθέτει την ομοιότητα μεταξύ της τροχιάς και του  $TR_{query}$  για όλα τα αντίστοιχα χρονικά διαστήματα στο  $T$  κανονικοποιώντας στο τέλος το αποτέλεσμα. Οι ομοιότητες που υπολογίστηκαν ταξινομούνται με φθίνουσα σειρά κι επιστρέφονται οι  $k$  τροχιές με τις μεγαλύτερες ομοιότητες.

Ακολουθεί ο ψευδοκώδικας για το βασικό αλγόριθμο:

## Βασικός Αλγόριθμος

---

ΕΙΣΟΔΟΣ: (1) Μια τροχιά  $TR_{query}$  και ένα χρονικό παράθυρο  $T = \{t_1, t_2, \dots, t_n\}$   
(2) Ένα σύνολο τροχιών  $\mathcal{T} = \{TR_1, \dots, TR_n\}$   
(3) Ο αριθμός  $k$  των κοντινότερων γειτόνων προς εύρεση

ΕΞΟΔΟΣ:  $(TR_j, \text{κατάταξη } m)$ , αν  $TR_j \in \mathcal{T}$  είναι ο  $m$ -οστός κοντινότερος γείτονας, όπου  $m \leq k$

ΑΛΓΟΡΙΘΜΟΣ:

1.  $S \leftarrow \{\}$
2. **for each** ( $TR \in \mathcal{T}$ ) **do**
3. 
$$sim_{TR} \leftarrow \frac{\sum_{t_i=t_1}^{t_n-1} sim_{t_i, t_{i+1}}(TR_{query}, TR)}{|T|-1}$$
4.  $S \leftarrow sim_{TR}$
5. ταξινόμηση του  $S$  με φθίνουσα σειρά με βάση το  $sim_{TR}$
6. προβολή των  $k$  τροχιών με τη μεγαλύτερη ομοιότητα  $sim_{TR}$

### 4.2.2 Βασικός αλγόριθμος με ουρά προτεραιότητας

Ο βασικός αλγόριθμος εκτελεί εξαντλητική αναζήτηση για την εύρεση των  $k$  κοντινότερων γειτόνων. Για να επιταχυνθεί ο αλγόριθμος, μπορούμε να μειώσουμε το χώρο αναζήτησης λύσεων χρησιμοποιώντας μια ουρά προτεραιότητας [CLR+09]. Όμως, η επιτάχυνση του αλγορίθμου δε θα είναι ευθέως ανάλογη του ποσοστού κλαδέματος (pruning) των τμημάτων τροχιών εξαιτίας του χρόνου  $O(\log(\text{πλήθος κινούμενων αντικειμένων}))$  που απαιτείται για να εξαχθούν και να εισαχθούν στοιχεία στην ουρά προτεραιότητας.

Παρακάτω παρατίθεται ο ψευδοκώδικας για την επέκταση του βασικού αλγορίθμου με εισαγωγή μιας ουράς προτεραιότητας. Αρχικά η ουρά προτεραιότητας είναι άδεια και η συνολική ομοιότητα κάθε τροχιάς με το  $TR_{query}$  αρχικοποιείται στη μέγιστη δυνατή τιμή, δηλαδή ίση με 1 για κάθε χρονικό διάστημα στο παράθυρο  $T$ . Κατόπιν υπολογίζεται η ομοιότητα μεταξύ κάθε τροχιάς και του  $TR_{query}$  για το πρώτο χρονικό διάστημα  $[t_1, t_2]$  στο  $T$  και διορθώνεται η συνολική ομοιότητα στη γραμμή 4. Στη γραμμή 5 η συνολική ομοιότητα κάθε τροχιάς προστίθεται στην ουρά προτεραιότητας.

Έπειτα, όσο δεν έχουν βρεθεί και οι  $k$  κοντινότεροι γείτονες, ο αλγόριθμος εξάγει από την ουρά την τροχιά με τη μεγαλύτερη συνολική ομοιότητα. Εάν αυτή η ομοιότητα έχει διορθωθεί για κάθε χρονικό διάστημα στο  $T$ , τότε συμπεραίνουμε ότι η τροχιά ανήκει στους κοντινότερους γείτονες αφού έχει συνολική ομοιότητα μεγαλύτερη από κάθε άλλη τροχιά στην ουρά. Στο εξής, απομένει να βρεθεί ένας λιγότερος κοντινότερος γείτονας. Αλλιώς, η ομοιότητα μεταξύ της τροχιάς και του  $TR_{query}$  υπολογίζεται για το επόμενο χρονικό διάστημα στο  $T$ , η τιμή της συνολικής ομοιότητας διορθώνεται και προστίθεται στην ουρά.

### Βασικός αλγόριθμος με ουρά προτεραιότητας

---

ΕΙΣΟΔΟΣ: (1) Μια τροχιά  $TR_{query}$  και ένα χρονικό παράθυρο  $T = \{t_1, t_2, \dots, t_n\}$   
 (2) Ένα σύνολο τροχιών  $\mathcal{T} = \{TR_1, \dots, TR_n\}$   
 (3) Ο αριθμός  $k$  των κοντινότερων γειτόνων προς εύρεση  
 ΕΞΟΔΟΣ:  $(TR_j, \text{κατάταξη } m)$ , αν  $TR_j \in \mathcal{T}$  είναι ο  $m$ -οστός κοντινότερος γείτονας, όπου  $m \leq k$

ΑΛΓΟΡΙΘΜΟΣ:

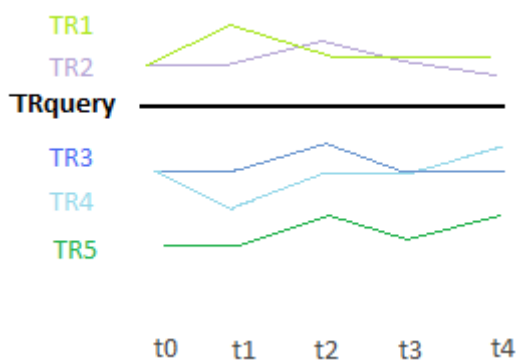
```

1. Queue ← NewPriorityQueue()
2. for each (TR ∈  $\mathcal{T}$ ) do
3.    $sim_{TR} \leftarrow |T| - 1$ 
4.    $sim_{TR} \leftarrow sim_{TR} - 1 + sim_{t_1, t_2}(TR_{query}, TR)$ 
5.   προσθήκη ( $sim_{TR}, t_2$ ) στην ουρά Queue
6. while (k > 0) do
7.   εξαγωγή μέγιστου στοιχείου ( $sim_{TR}, t_{next}$ ) από την ουρά Queue
8.   if  $t_{next} \notin T$  then
9.     προβολή (TR, k)
10.    k ← k - 1
11.  else
12.     $sim_{TR} \leftarrow sim_{TR} - 1 + sim_{t_{next}, t_{next+1}}(TR_{query}, TR)$ 
13.    προσθήκη ( $sim_{TR}, t_{next+1}$ ) στην ουρά Queue

```

### Παράδειγμα

Ας υποθέσουμε ότι θέλουμε να βρούμε τους 2 κοντινότερους γείτονες στην τροχιά  $TR_{query}$  του σχήματος 4.1. Στη συνέχεια, περιγράφονται τα βήματα του αλγορίθμου και τα στοιχεία στην ουρά προτεραιότητας.



Σχήμα 4.1: Παράδειγμα εύρεσης k-NN

	$\text{sim}(TR_{query}, TR1)$	$\text{sim}(TR_{query}, TR2)$	$\text{sim}(TR_{query}, TR3)$	$\text{sim}(TR_{query}, TR4)$	$\text{sim}(TR_{query}, TR5)$
$t_1 = [t_0, t_1]$	0.8	0.95	0.9	0.85	0.60
$t_2 = [t_1, t_2]$	0.8	0.85	0.95	0.85	0.75
$t_3 = [t_2, t_3]$	0.95	0.85	0.95	0.9	0.75
$t_4 = [t_3, t_4]$	0.95	0.96	0.88	0.95	0.75

$$k = 2, |T| - 1 = 4.$$

- (1) Εισαγωγή των TR1, TR2, TR3, TR4, TR5 στην ουρά.  
 Ουρά:  $\{ (3.95, (TR2, t_2)), (3.90, (TR3, t_2)), (3.85, (TR4, t_2)), (3.80, (TR1, t_2)), (3.60, (TR5, t_2)) \}$ .
- (2) Εξαγωγή του TR2 από την ουρά.  
 $\text{sim}_{TR2} = 3.95 - 1 + 0.85 = 3.80$ .  
 Εισαγωγή του TR2 στην ουρά.  
 Ουρά:  $\{ (3.90, (TR3, t_2)), (3.85, (TR4, t_2)), (3.80, (TR2, t_3)), (3.80, (TR1, t_2)), (3.60, (TR5, t_2)) \}$ .
- (3) Εξαγωγή του TR3 από την ουρά.  
 $\text{sim}_{TR3} = 3.90 - 1 + 0.95 = 3.85$ .  
 Εισαγωγή του TR3 στην ουρά.  
 Ουρά:  $\{ (3.85, (TR3, t_3)), (3.85, (TR4, t_2)), (3.80, (TR2, t_3)), (3.80, (TR1, t_2)), (3.60, (TR5, t_2)) \}$ .



- (4) Εξαγωγή του TR3 από την ουρά.  
 $\text{sim}_{\text{TR3}} = 3.85 - 1 + 0.95 = 3.80$ .  
 Εισαγωγή του TR3 στην ουρά.  
 Ουρά: { ( 3.85 , (TR4, t<sub>2</sub>) ), ( 3.80 , (TR3, t<sub>4</sub>) ), ( 3.80 , (TR2, t<sub>3</sub>) ), ( 3.80 , (TR1, t<sub>2</sub>) ), ( 3.60 , (TR5, t<sub>2</sub>) ) }.
- (5) Εξαγωγή του TR4 από την ουρά.  
 $\text{sim}_{\text{TR4}} = 3.85 - 1 + 0.85 = 3.70$ .  
 Εισαγωγή του TR4 στην ουρά.  
 Ουρά: { ( 3.80 , (TR3, t<sub>4</sub>) ), ( 3.80 , (TR2, t<sub>3</sub>) ), ( 3.80 , (TR1, t<sub>2</sub>) ), ( 3.70 , (TR4, t<sub>3</sub>) ), ( 3.60 , (TR5, t<sub>2</sub>) ) }.
- (6) Εξαγωγή του TR3 από την ουρά.  
 $\text{sim}_{\text{TR3}} = 3.80 - 1 + 0.88 = 3.68$ .  
 Εισαγωγή του TR3 στην ουρά.  
 Ουρά: { ( 3.80 , (TR2, t<sub>3</sub>) ), ( 3.80 , (TR1, t<sub>2</sub>) ), ( 3.70 , (TR4, t<sub>3</sub>) ), ( 3.68 , (TR3, t<sub>5</sub>) ), ( 3.60 , (TR5, t<sub>2</sub>) ) }.
- (7) Εξαγωγή του TR2 από την ουρά.  
 $\text{sim}_{\text{TR4}} = 3.80 - 1 + 0.85 = 3.65$ .  
 Εισαγωγή του TR2 στην ουρά.  
 Ουρά: { ( 3.80 , (TR1, t<sub>2</sub>) ), ( 3.70 , (TR4, t<sub>3</sub>) ), ( 3.68 , (TR3, t<sub>5</sub>) ), ( 3.65 , (TR2, t<sub>4</sub>) ), ( 3.60 , (TR5, t<sub>2</sub>) ) }.
- (8) Εξαγωγή του TR1 από την ουρά.  
 $\text{sim}_{\text{TR1}} = 3.80 - 1 + 0.8 = 3.60$ .  
 Εισαγωγή του TR1 στην ουρά.  
 Ουρά: { ( 3.70 , (TR4, t<sub>3</sub>) ), ( 3.68 , (TR3, t<sub>5</sub>) ), ( 3.65 , (TR2, t<sub>4</sub>) ), ( 3.60 , (TR1, t<sub>3</sub>) ), ( 3.60 , (TR5, t<sub>2</sub>) ) }.
- (9) Εξαγωγή του TR4 από την ουρά.  
 $\text{sim}_{\text{TR4}} = 3.70 - 1 + 0.9 = 3.60$ .  
 Εισαγωγή του TR4 στην ουρά.  
 Ουρά: { ( 3.68 , (TR3, t<sub>5</sub>) ), ( 3.65 , (TR2, t<sub>4</sub>) ), ( 3.60 , (TR4, t<sub>4</sub>) ), ( 3.60 , (TR1, t<sub>3</sub>) ), ( 3.60 , (TR5, t<sub>2</sub>) ) }.
- (10) Εξαγωγή του TR3 από την ουρά. k = 1.  
 Προβολή του TR3 ως 1<sup>ο</sup> κοντινότερο γείτονα.  
 Ουρά: { ( 3.65 , (TR2, t<sub>4</sub>) ), ( 3.60 , (TR4, t<sub>4</sub>) ), ( 3.60 , (TR1, t<sub>3</sub>) ), ( 3.60 , (TR5, t<sub>2</sub>) ) }.
- (11) Εξαγωγή του TR2 από την ουρά.  
 $\text{sim}_{\text{TR2}} = 3.65 - 1 + 0.96 = 3.61$ .  
 Εισαγωγή του TR2 στην ουρά.  
 Ουρά: { ( 3.61 , (TR2, t<sub>5</sub>) ), ( 3.60 , (TR4, t<sub>4</sub>) ), ( 3.60 , (TR1, t<sub>3</sub>) ), ( 3.60 , (TR5, t<sub>2</sub>) ) }.
- (12) Εξαγωγή του TR2 από την ουρά. k = 0.  
 Προβολή του TR2 ως 2<sup>ο</sup> κοντινότερο γείτονα.

### 4.2.3 Προσεγγιστικός αλγόριθμος

Ένας εναλλακτικός τρόπος επιτάχυνσης του βασικού αλγορίθμου είναι να γίνει προσέγγιση της τροχιάς χρησιμοποιώντας μια μέθοδο απλοποίησης γραμμής.

Παρακάτω δίνεται ο ψευδοκώδικας του προσεγγιστικού αλγορίθμου. Εκτός από την παράμετρο  $k$ , χρειάζεται να προσδιοριστούν και δύο παράμετροι ανοχής (για την απλοποίηση της τροχιάς). Αρχικά, ο αλγόριθμος απλοποιεί την τροχιά του  $TR_{query}$ . Έπειτα, εξετάζει επαναληπτικά όλες τις τροχιές στο  $\mathcal{T}$ , τις απλοποιεί, προσθέτει τις ομοιότητες με βάρη μεταξύ της τρέχουσας απλοποιημένης τροχιάς και της απλοποιημένης τροχιάς του  $TR_{query}$  στα αντίστοιχα χρονικά διαστήματα στο  $T$  και τέλος κανονικοποιεί το αποτέλεσμα. Λόγω της απλοποίησης τροχιάς, τα χρονικά διαστήματα των τμημάτων των τροχιών μπορεί να είναι άνισα. Σε αυτή την περίπτωση, θεωρούμε διαδοχικά την τομή των διαστημάτων και τα αντίστοιχα τμήματα της τροχιάς. Αν απαιτείται ένα ενδιάμεσο σημείο από την απλοποιημένη τροχιά, χρησιμοποιείται η μέθοδος γραμμικής παρεμβολής για να βρεθεί.

Τέλος, οι ομοιότητες που έχουν υπολογιστεί ταξινομούνται με φθίνουσα σειρά και επιστρέφονται οι  $k$  τροχιές με τη μεγαλύτερη ομοιότητα.

---

#### Προσεγγιστικός Αλγόριθμος

- ΕΙΣΟΔΟΣ: (1) Μια τροχιά  $TR_{query}$  και ένα χρονικό παράθυρο  $T = \{t_1, t_2, \dots, t_n\}$   
 (2) Ένα σύνολο τροχιών  $\mathcal{T} = \{TR_1, \dots, TR_n\}$   
 (3) Ο αριθμός  $k$  των κοντινότερων γειτόνων προς εύρεση  
 (4) Οι παράμετροι ανοχής απόστασης  $\epsilon$  & ανοχής ταχύτητας  $\delta$

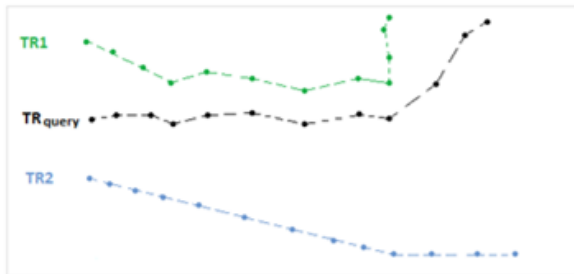
ΕΞΟΔΟΣ:  $(TR_j, \text{κατάταξη } m)$ , αν  $TR_j \in \mathcal{T}$  είναι ο  $m$ -οστός κοντινότερος γείτονας, όπου  $m \leq k$

ΑΛΓΟΡΙΘΜΟΣ:

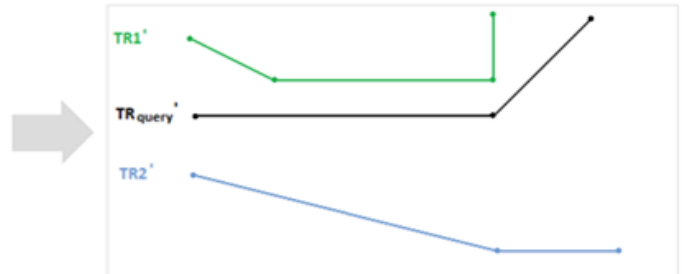
1.  $S \leftarrow \{\}$
2.  $TR_{query}' \leftarrow \text{Zhao - Saalfeld extended line simplification}(TR_{query}, \epsilon, \delta)$
3. **for each**  $(TR \in \mathcal{T})$  **do**
4.      $TR' \leftarrow \text{Zhao - Saalfeld extended line simplification}(TR, \epsilon, \delta)$
5.      $sim_{TR} \leftarrow \frac{\sum_{t'_i=t_1}^{t_{n-1}'} w_i \cdot sim_{t'_i t'_{i+1}}(TR_{query}', TR')}{|T|-1}$ ,

όπου  $w_i = \#t_i \in [t_i', t_{i+1}')$

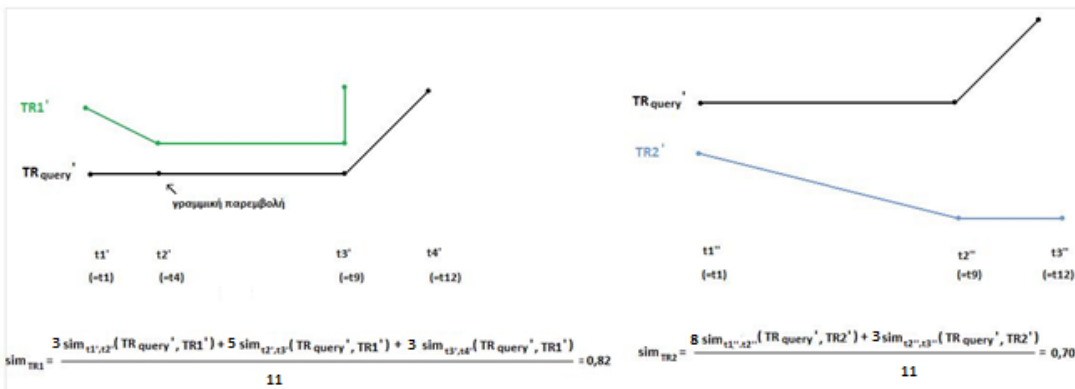
6.  $S \leftarrow \text{sim}_{\text{TR}}$
7. ταξινόμηση του  $S$  με φθίνουσα σειρά με βάση το  $\text{sim}_{\text{TR}}$
8. προβολή των  $k$  τροχιών με τη μεγαλύτερη ομοιότητα  $\text{sim}_{\text{TR}}$



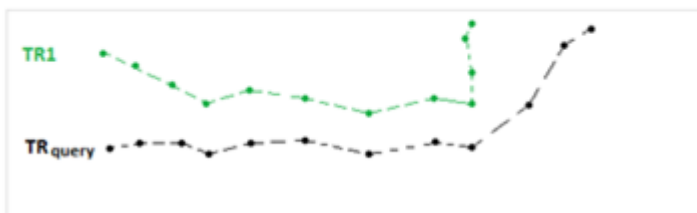
Αρχικές τροχιές



Βήμα 1: Απλοποίηση τροχιών



Βήμα 2: Υπολογισμός ομοιοτήτων

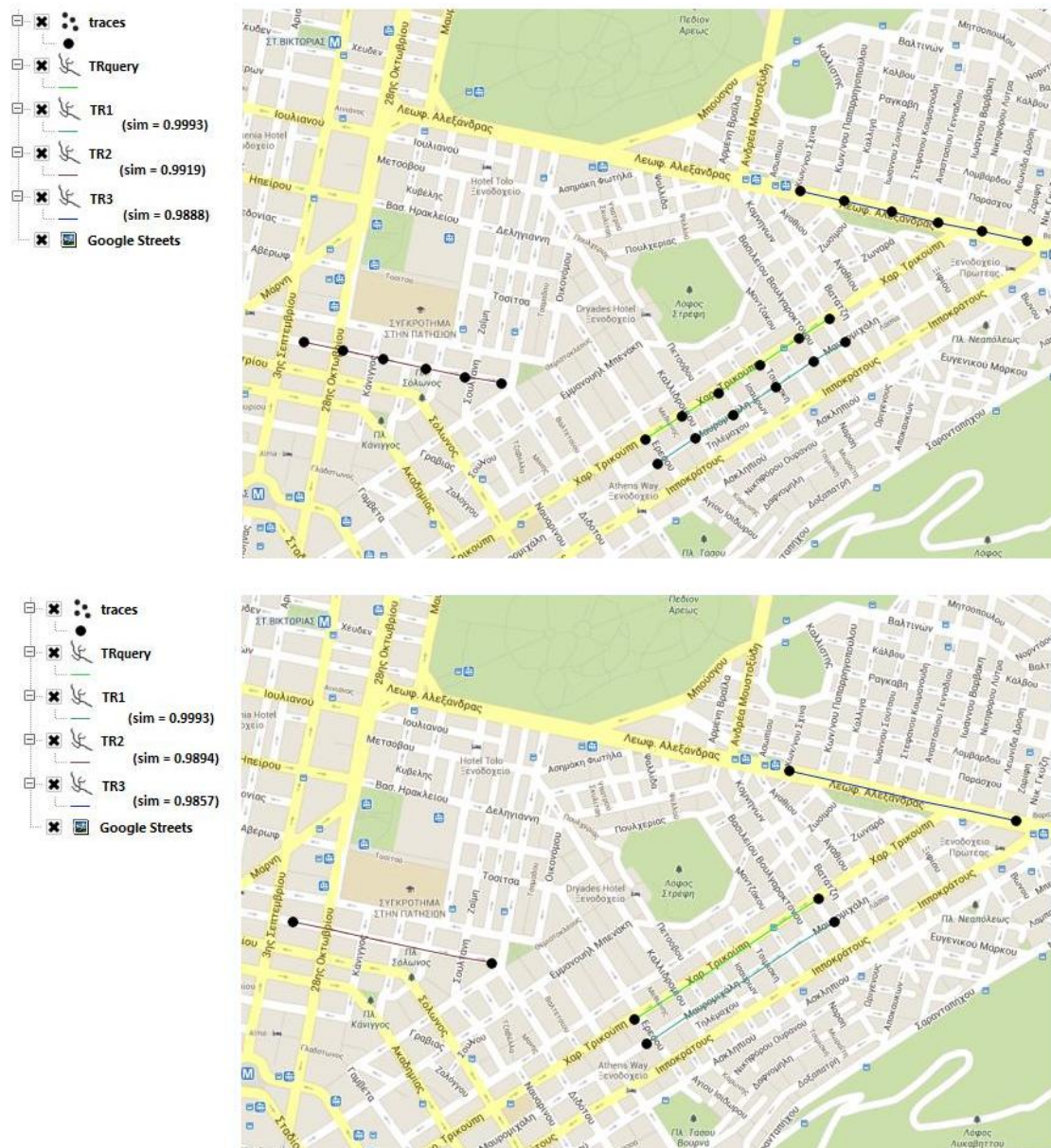


Βήμα 3: Επιλογή  $k$  κοντινότερων γειτόνων ( $k=1$ )

Σχήμα 4.2: Παράδειγμα εκτέλεσης προσεγγιστικού αλγόριθμου

Ο προσεγγιστικός αλγόριθμος εκτελεί λιγότερους υπολογισμούς ομοιότητας, βελτιώνοντας το χρόνο εκτέλεσης αλλά μειώνοντας επίσης την ακρίβεια των αποτελεσμάτων. Η ακρίβεια περιορίζεται για τους εξής λόγους:

Η ομοιότητα υπολογίζεται με βάση ευθύγραμμα τμήματα που συνδέουν μη διαδοχικά σημεία της αρχικής τροχιάς. Κατά συνέπεια, ο υπολογισμός ομοιότητας είναι προσεγγιστικός, ακόμα και όταν οι παράμετροι ανοχής τίθενται ίσοι με το μηδέν.



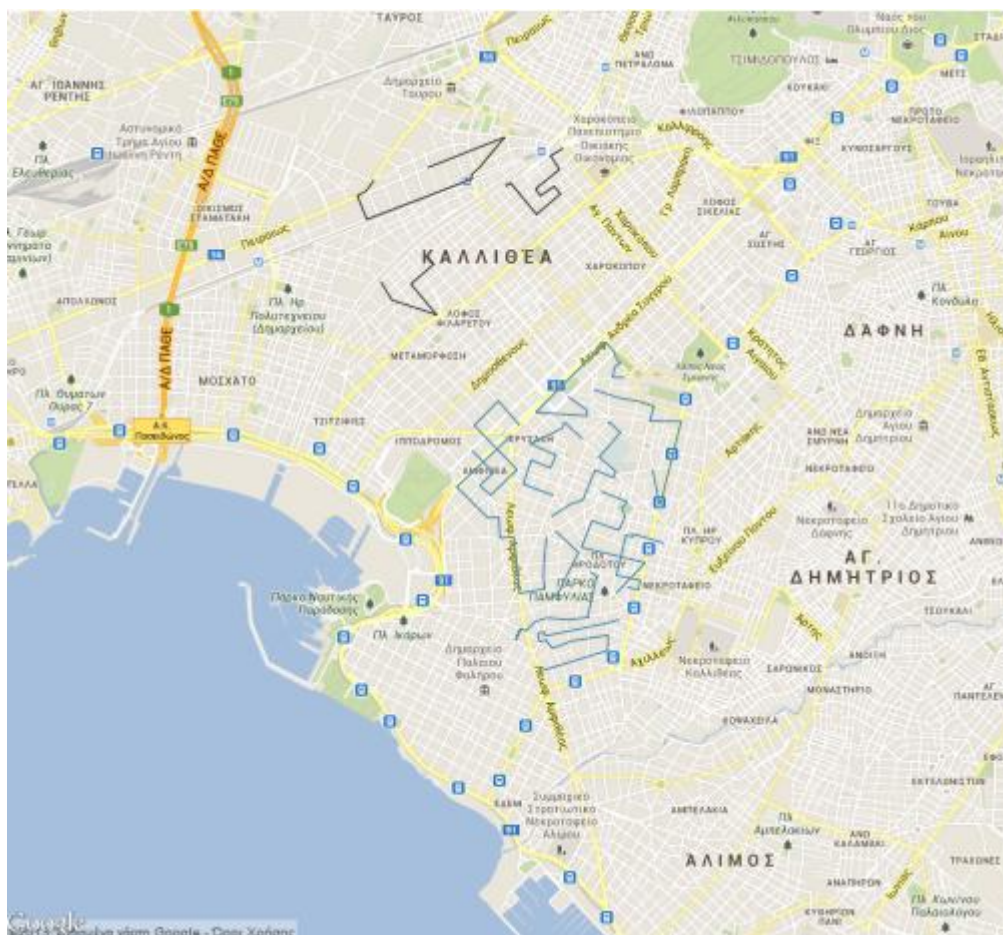
Σχήμα 4.3: Παράδειγμα προσεγγιστικής εύρεσης της ομοιότητας (με μηδενικές παραμέτρους ανοχής)

- Λόγω των παραμέτρων ανοχής, η ομοιότητα υπολογίζεται για μια προσεγγιστική αναπαράσταση της τροχιάς.



Σχήμα 4.4: Παράδειγμα προσεγγιστικής εύρεσης της ομοιότητας (με μη μηδενικές παραμέτρους ανοχής)

Επειδή η τιμή ομοιότητας διαφέρει από την αντίστοιχη του βασικού αλγορίθμου, αλλά δεν επηρεάζεται ομοιόμορφα για όλες τις τροχιές, η προσεγγιστική λίστα των κ κοντινότερων γειτόνων μπορεί να διαφέρει από την πραγματική. Το ποσοστό διαφοράς είναι διαφορετικό για κάθε σύνολο δεδομένων κι εξαρτάται από παράγοντες, όπως η πυκνότητα των τροχιών ανά περιοχή και η ομαλότητα των τροχιών. Για παράδειγμα, στο Σχήμα 4.5 αναμένουμε ότι στην περιοχή της Νέας Σμύρνης ο κοντινότερος γείτονας σύμφωνα με τον προσεγγιστικό αλγόριθμο θα μεταβάλλεται πολύ συχνά όσο αυξάνονται οι τιμές των παραμέτρων ανοχής. Αντίθετα, στην Καλλιθέα, που δεν υπάρχουν πολλές τροχιές, ο κοντινότερος γείτονας δε θα επηρεάζεται εύκολα από τις μεταβολές των παραμέτρων ανοχής.



Σχήμα 4.5: Περιοχές με διαφορετική πυκνότητα τροχιών

Διαισθητικά αναμένεται ότι μικρές τιμές των παραμέτρων ανοχής θα παρέχουν ικανοποιητική ακρίβεια στα αποτελέσματα της προσεγγιστικής μεθόδου. Ωστόσο, λόγω της σημαντικής εξάρτησης της ακρίβειας από εγγενή χαρακτηριστικά του συνόλου δεδομένων, είναι δύσκολο να καθορίσουμε το ποσοστό ακρίβειας που επιτυγχάνεται για δοθείσες τιμές των παραμέτρων ανοχής ή αντίστροφα τις κατάλληλες τιμές ανοχής για να επιτευχθεί το επιθυμητό ποσοστό ακρίβειας.





# 5

## Πειραματική ανάλυση και αξιολόγηση

Σε αυτό το κεφάλαιο αξιολογούμε την επίδοση των αλγορίθμων που αναπτύχθηκαν. Αρχικά περιγράφουμε τα πραγματικά σύνολα δεδομένων που χρησιμοποιήθηκαν για τα πειράματα και το υπολογιστικό περιβάλλον εκτέλεσης. Στη συνέχεια, παρουσιάζουμε τα αποτελέσματα ως προς το χρόνο εκτέλεσης και την ακρίβεια.

### 5.1 Σύνολα δεδομένων

Τα πειράματα πραγματοποιήθηκαν σε τρία πραγματικά σύνολα δεδομένων.

- **T-Drive trajectory dataset<sup>1</sup>**

Αυτό το σύνολο δεδομένων αποτελεί μέρος του T-Drive trajectory dataset [YZX+11, YZZ+10] και περιλαμβάνει τροχιές από 569 ταξί στην περιοχή του Πεκίνου κατά την περίοδο 2/2/2008 - 4/2/2008. Μετά την προεπεξεργασία των δεδομένων, ο αριθμός των τροχιών ανέρχεται σε 1.023.924.

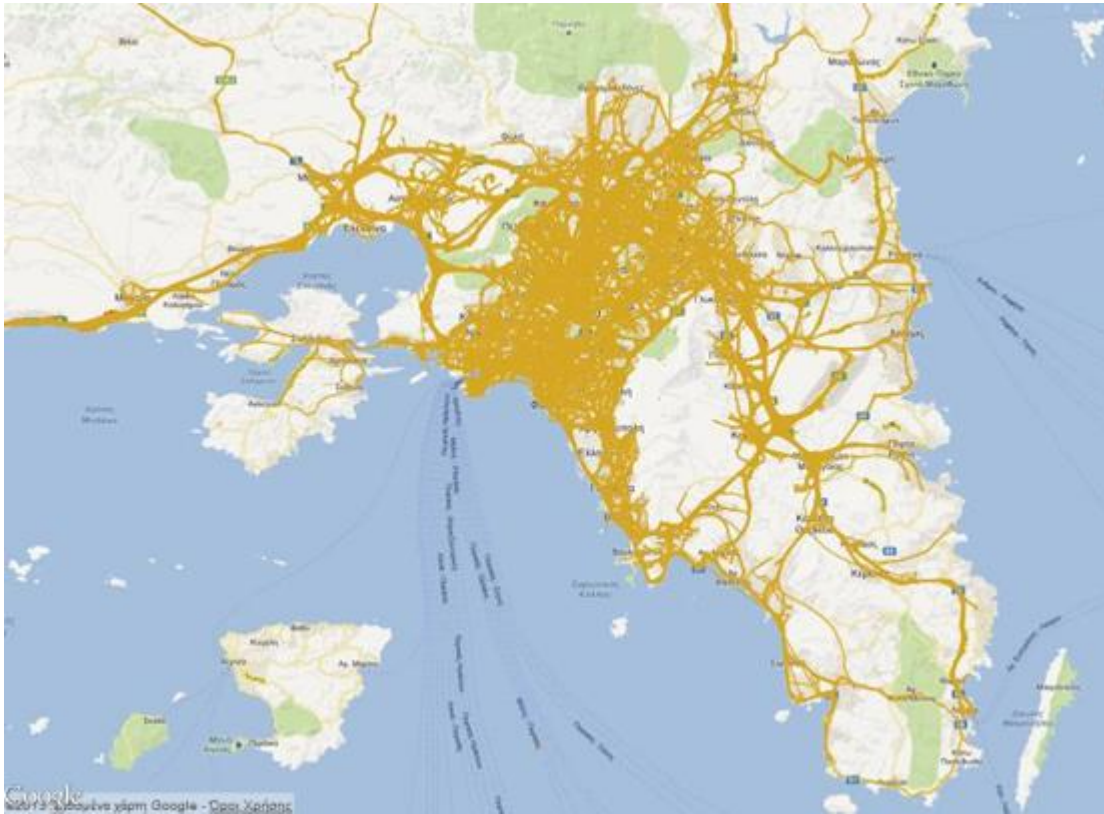
---

<sup>1</sup><http://research.microsoft.com/apps/pubs/?id=152883>



- **FCD Athens trajectory dataset<sup>3</sup>**

Αυτό το σύνολο δεδομένων περιέχει 667.421 τροχιές από 2497 διαφορετικά οχήματα εξοπλισμένα με GPS. Τα οχήματα κινούνταν στο νομό Αττικής και η καταγραφή των θέσεων τους έγινε για μια ημέρα (01/10/2012).



**Σχήμα 5.3:** Οπτικοποίηση τροχιών από το FCD Athens trajectory dataset

Για τα παραπάνω σύνολα δεδομένων χρησιμοποιήθηκε περίοδος δειγματοληψίας ίση με 30 δευτερόλεπτα κατά την προεπεξεργασία.

---

<sup>3</sup>Το σύνολο δεδομένων FCD Athens trajectory dataset παραχωρήθηκε από το Ινστιτούτο Πληροφοριακών Συστημάτων (ΠΠΣΥ) του Ερευνητικού Κέντρου Αθηνά.

## 5.2 Περιβάλλον εκτέλεσης και κριτήρια αξιολόγησης

Οι αλγόριθμοι υλοποιήθηκαν σε Java και τα πειράματα διεξήχθησαν σε μηχανήμα με τέσσερις επεξεργαστές 2.1 GHz και μνήμη 2GB. Στο μηχανήμα τρέχει λειτουργικό Ubuntu 12.04 LTS. Για τα σύνολα δεδομένων χρησιμοποιήθηκε βάση δεδομένων PostgreSQL 9.1<sup>4</sup> με PostGIS επέκταση. Η PostGIS<sup>5</sup> επέκταση παρέχει υποστήριξη για γεωγραφικά αντικείμενα και καλύτερη διαχείριση των χωρικών βάσεων δεδομένων. Η οπτικοποίηση των αποτελεσμάτων έγινε με την πλατφόρμα Quantum GIS<sup>6</sup>.

Για να αξιολογήσουμε την ακρίβεια των αποτελεσμάτων, συγκρίναμε τη λίστα L των k κοντινότερων γειτόνων των άλλων μεθόδων με την αντίστοιχη λίστα L<sub>0</sub> του βασικού αλγόριθμου με βάση την απόσταση Spearman F\* [FKS03]:

$$F^*(L, L_0) = \sum_{i \in D} |\text{pos}(i, L) - \text{pos}(i, L_0)| / \max F^*,$$

όπου D είναι το σύνολο των k κοντινότερων γειτόνων στη λίστα L<sub>0</sub>

pos(i,L) η συνάρτηση που επιστρέφει τη θέση του i στη λίστα L αν  $i \in L$  ή  $|L|+1$

και maxF\* η μέγιστη τιμή που μπορεί να πάρει η F\*, δηλαδή (k+1) x k.

## 5.3 Αποτελέσματα

Κατά τη διεξαγωγή των πειραμάτων συγκρίνουμε την επίδοση των αλγορίθμων κι εξετάζουμε πώς επηρεάζεται η ακρίβεια και ο χρόνος εκτέλεσης τους ανάλογα με:

- τον αριθμό k των κοντινότερων γειτόνων προς εύρεση
- τη διάρκεια του χρονικού παραθύρου αναζήτησης
- τις παραμέτρους ανοχής, στην περίπτωση του προσεγγιστικού αλγόριθμου

Συγκεκριμένα, επιλέχθηκαν 100 τυχαίες τροχιές και χρονικά παράθυρα δύο ωρών. Εντός των συγκεκριμένων χρονικών παραθύρων διεξήχθησαν πειράματα για υποδιαστήματα διαφορετικής διάρκειας, για διαφορετικές τιμές κοντινότερων γειτόνων k, καθώς και για διαφορετικές τιμές παραμέτρων ανοχής του προσεγγιστικού αλγόριθμου.

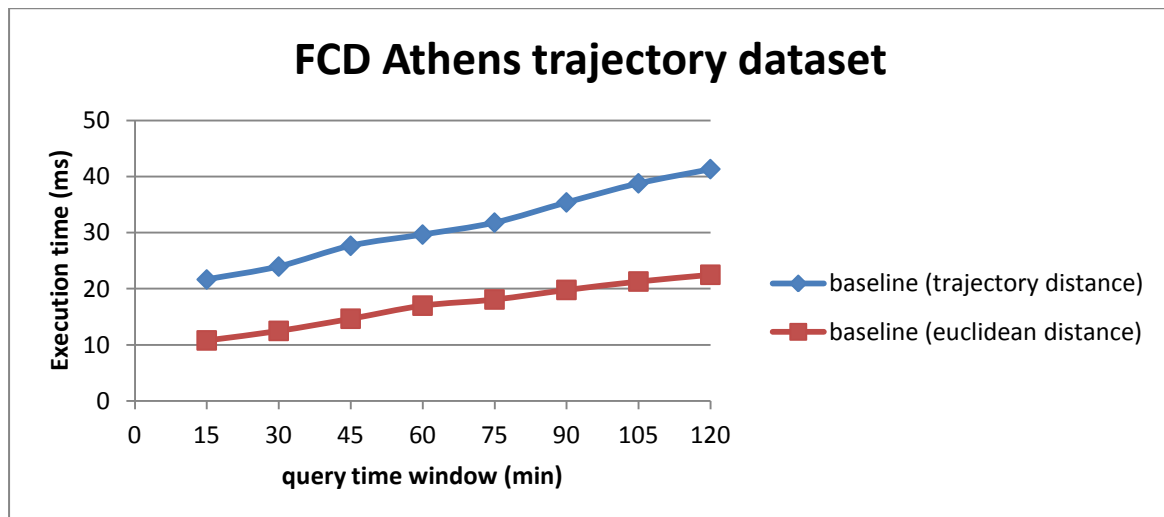
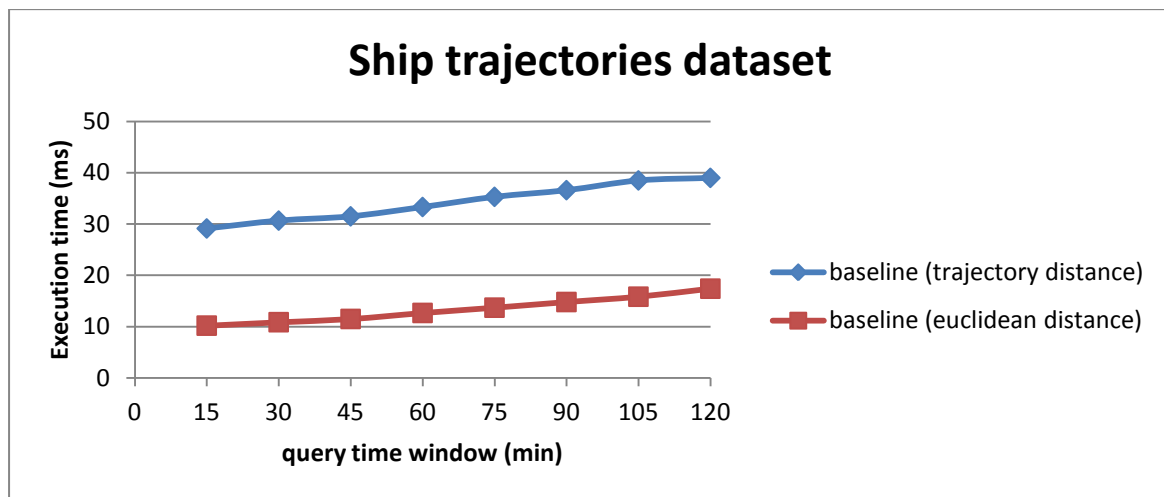
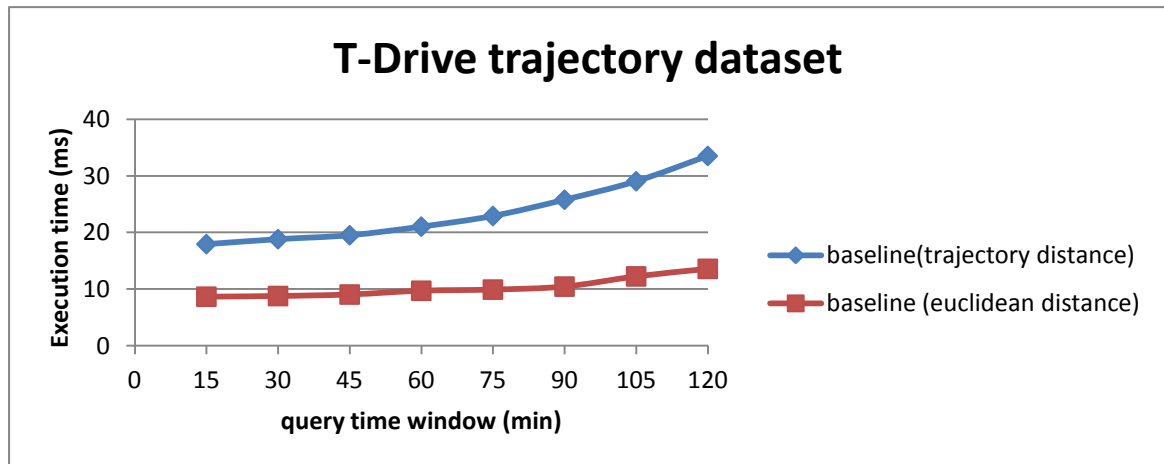
---

<sup>4</sup><http://www.postgresql.org/>

<sup>5</sup><http://postgis.net/>

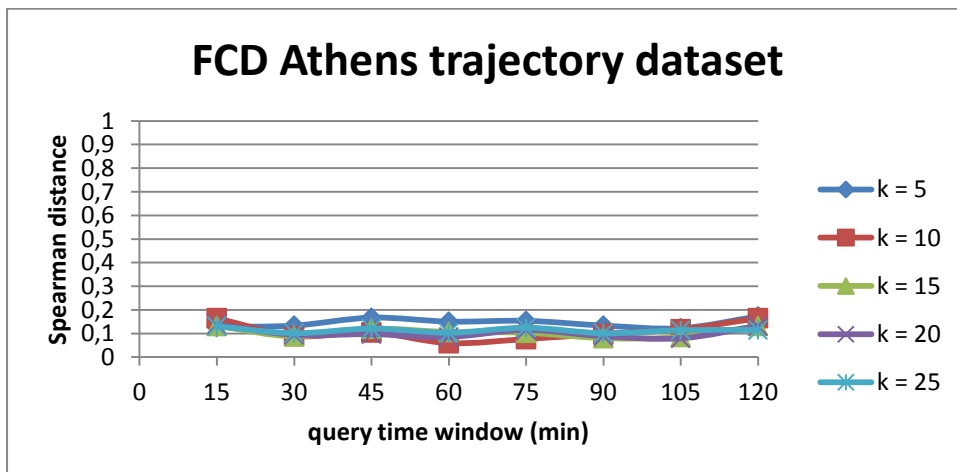
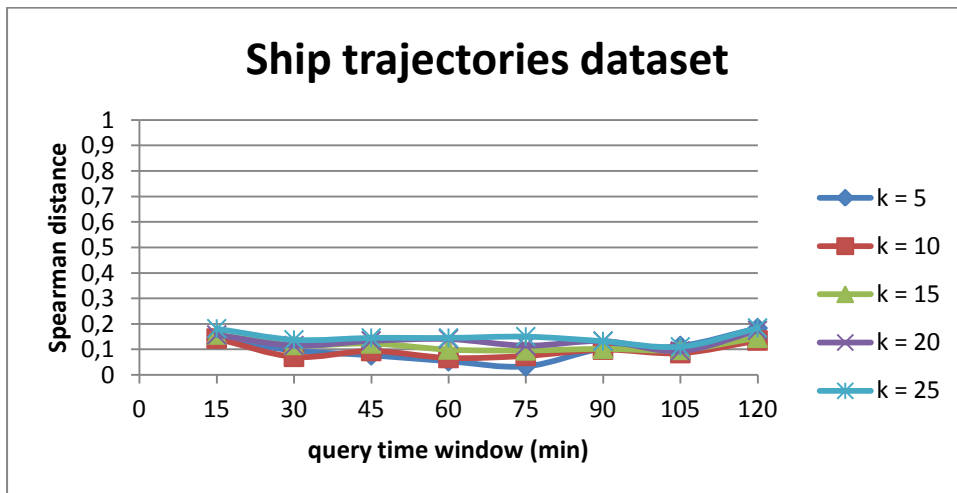
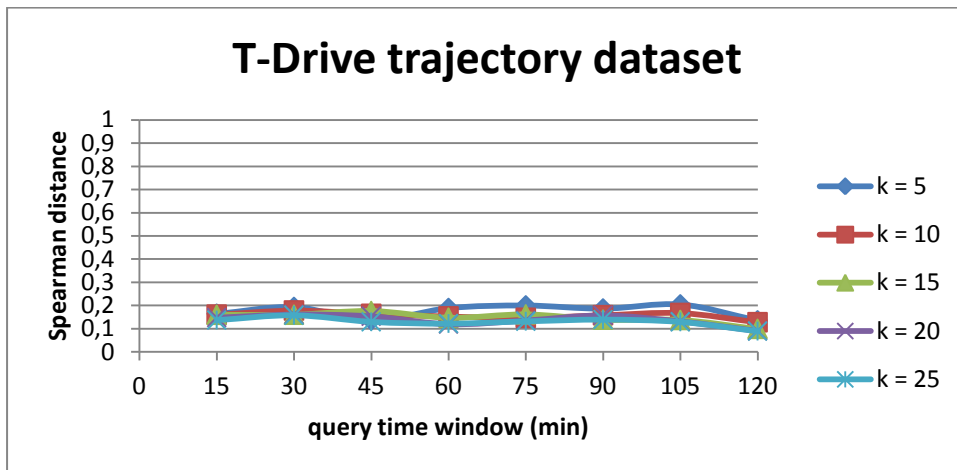
<sup>6</sup><http://www.qgis.org/>

## Σύγκριση της Hausdorff Απόστασης τροχιάς και της Ευκλείδειας Απόστασης



Παρατηρούμε ότι η χρήση της Hausdorff Απόστασης τροχιάς εισάγει πιο περίπλοκους υπολογισμούς με αποτέλεσμα να επιβραδύνεται η εκτέλεση του βασικού αλγορίθμου και να απαιτεί περίπου διπλάσιο χρόνο.

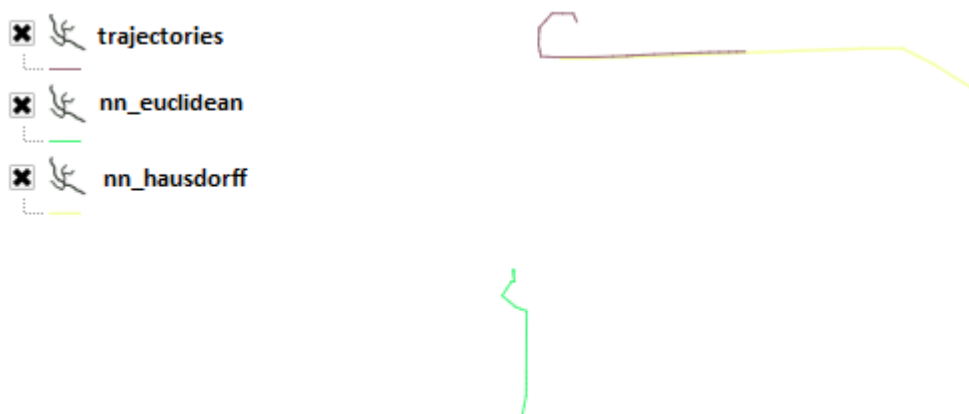
Όπως παρατηρείται στα επόμενα διαγράμματα, η λίστα των  $k$  κοντινότερων γειτόνων που βρίσκουμε χρησιμοποιώντας τη Hausdorff Απόσταση τροχιάς διαφέρει από τη λίστα που βρίσκουμε χρησιμοποιώντας την Ευκλείδεια Απόσταση. Συγκεκριμένα, τα αποτελέσματα παρουσιάζουν κατά μέσο όρο απόσταση Spearman που κυμαίνεται μεταξύ 0,1 και 0,2 αλλά κατά τη διάρκεια των πειραμάτων εμφανίστηκαν τιμές απόστασης έως και 1 σε ορισμένα ερωτήματα, δηλαδή όλοι οι κοντινότεροι γείτονες βρέθηκαν διαφορετικοί.



Πιο αξιόπιστη θεωρείται η χρήση της Hausdorff Απόστασης τροχιάς, καθώς συγκρίνει ολόκληρα τμήματα της τροχιάς αντί για μεμονωμένα σημεία. Αυτό επιβεβαιώνεται και από την οπτικοποίηση των αποτελεσμάτων. Ενδεικτικά, παρατίθενται δύο παραδείγματα οπτικοποίησης από το T-Drive trajectory dataset.

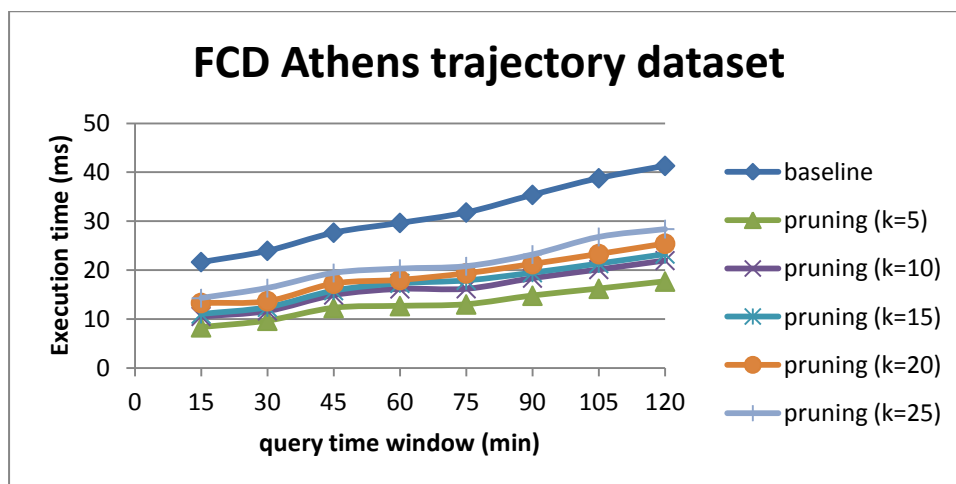
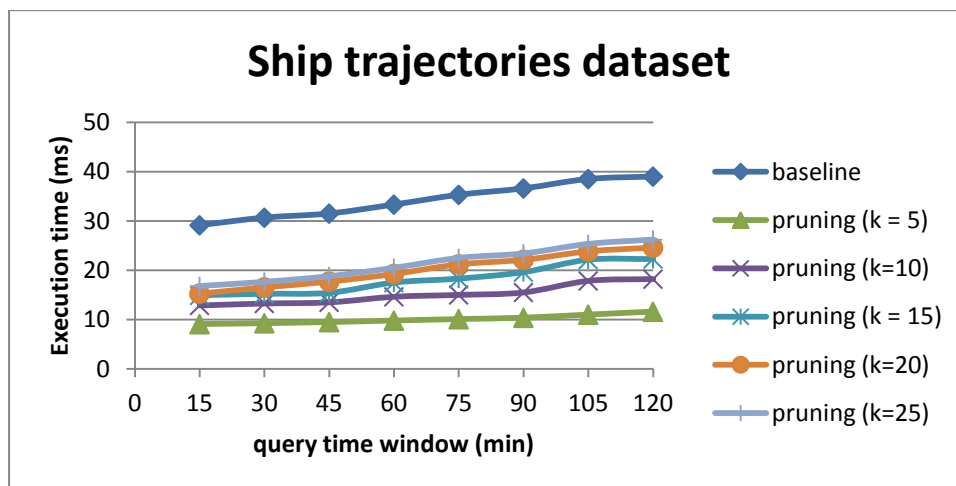
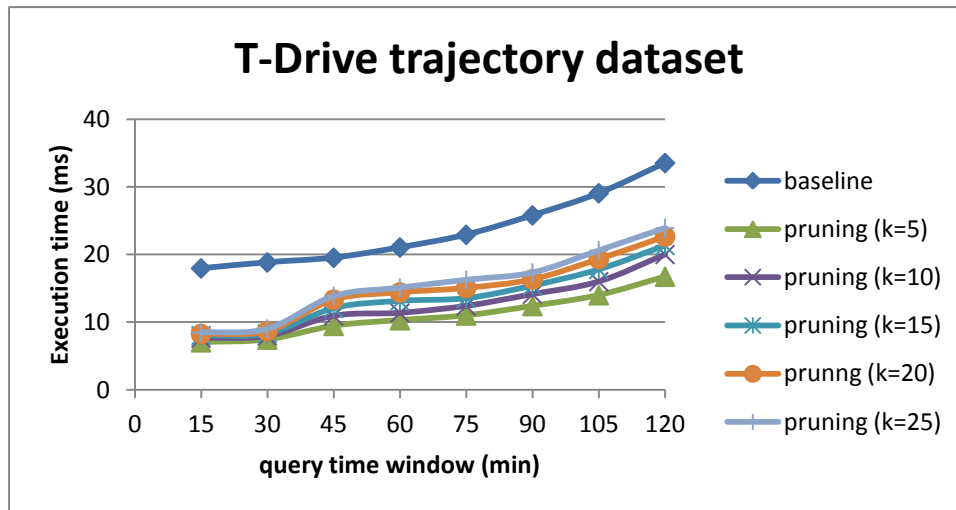


**Σχήμα 5.4:** Παράδειγμα κοντινότερου γείτονα με ευκλείδεια και hausdorff απόσταση τροχιάς. Χρησιμοποιήθηκε χρονικό παράθυρο  $T = [2008-02-03 19:40:36.604, 2008-02-03 19:55:36.604]$  και τροχιά ερώτησης TRquery με id 8838 από το T-Drive trajectory dataset.



**Σχήμα 5.5:** Παραδείγματα κοντινότερου γείτονα με ευκλείδεια και hausdorff απόσταση τροχιάς. Χρησιμοποιήθηκε χρονικό παράθυρο  $T = [2008-02-03 20:37:14.000, 2008-02-03 20:52:36.604]$  και τροχιά ερώτησης TRquery = 8179 από το T-Drive trajectory dataset.

## Σύγκριση της βασικής μεθόδου και της μεθόδου με κλάδεμα

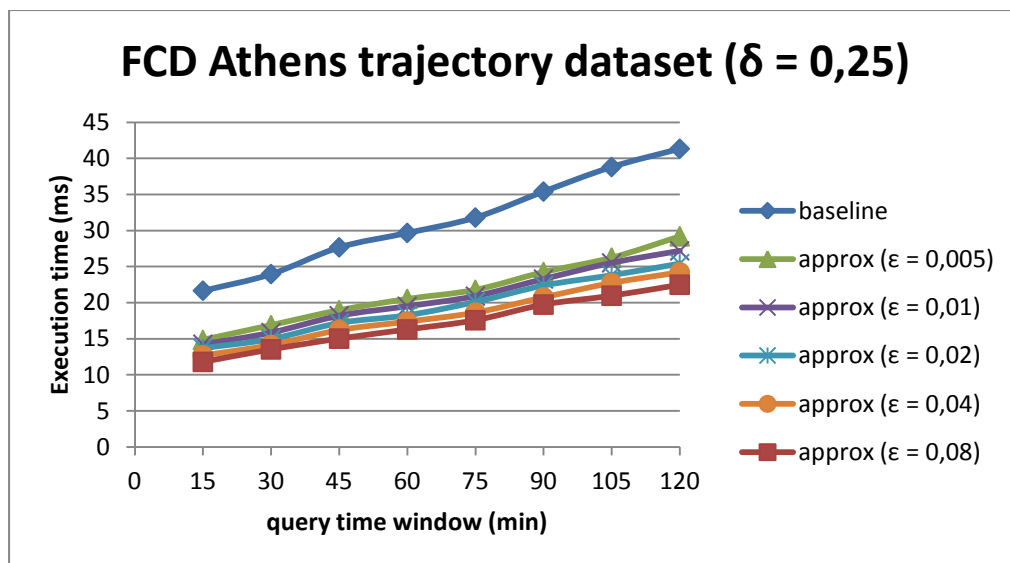
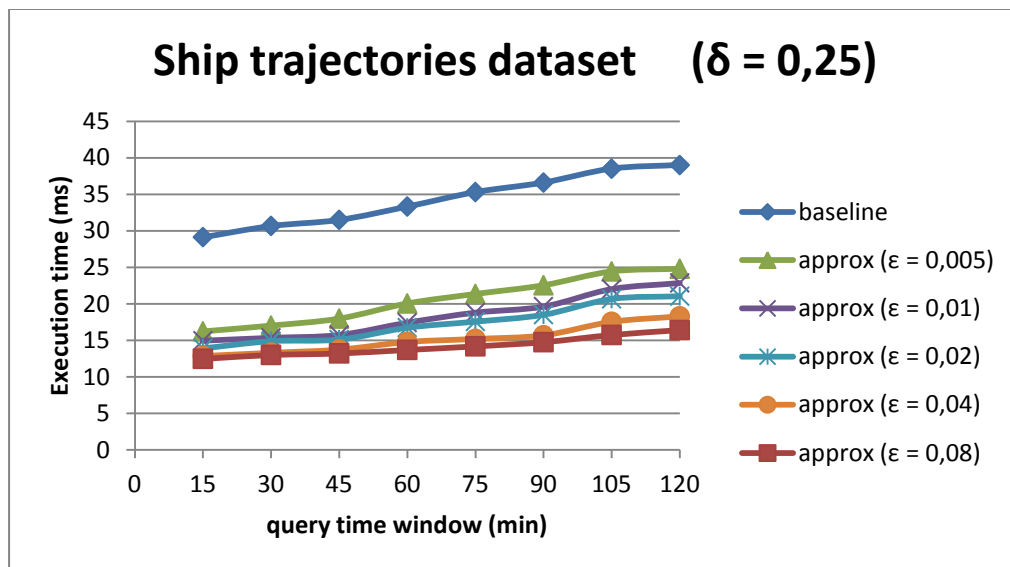
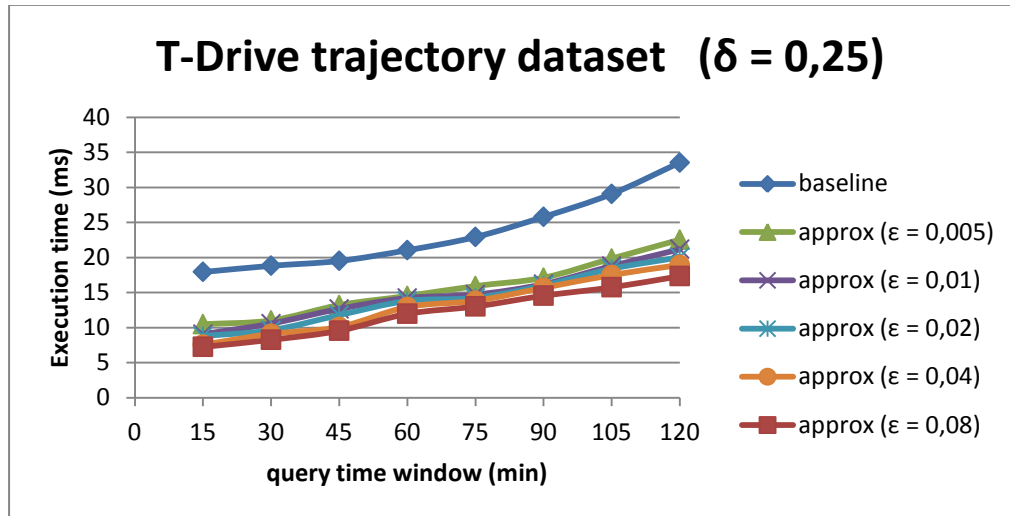


Παρατηρούμε ότι ο χρόνος εκτέλεσης βελτιώνεται σημαντικά με την προσθήκη ουράς προτεραιότητας. Αυξάνοντας τον αριθμό κοντινότερων γειτόνων που αναζητούμε (δηλαδή το  $k$ ) σε ένα ερώτημα, μειώνεται το κλάδεμα του χώρου αναζήτησης λύσεων και κατ' επέκταση αυξάνει ο χρόνος εκτέλεσης. Όμως η αύξηση του χρόνου εκτέλεσης δεν είναι γραμμική, όπως παρατηρείται από τα διαγράμματα, αφού εξαρτάται από το ποσοστό κλαδέματος.

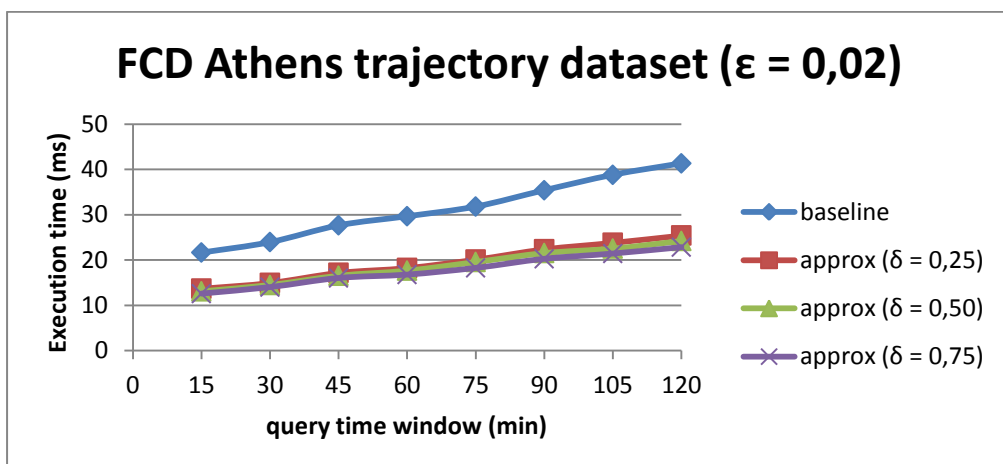
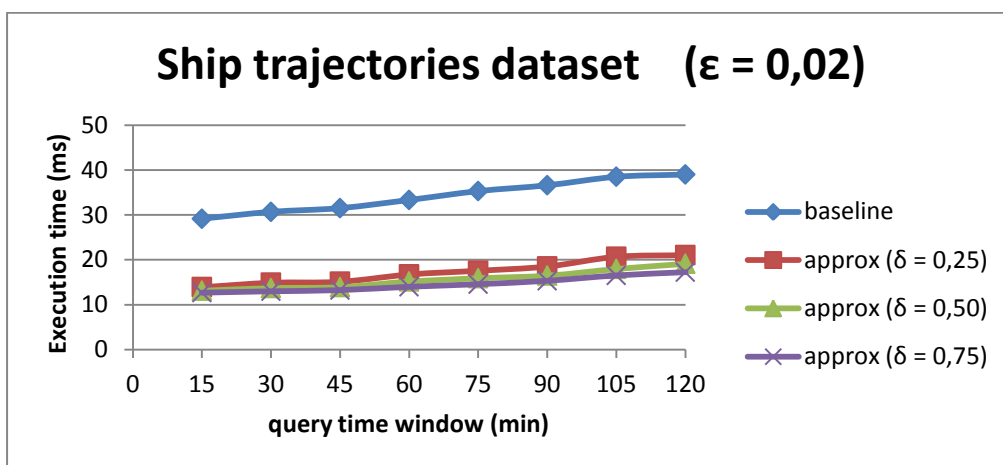
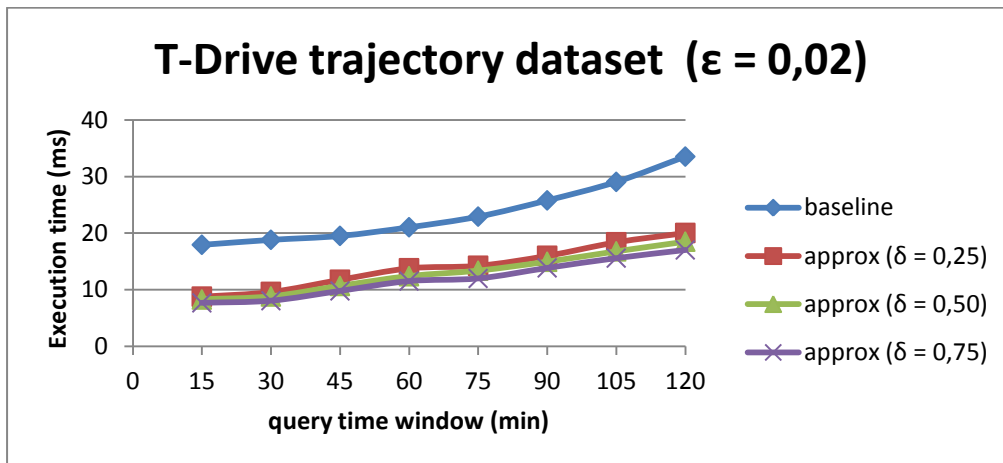


## Σύγκριση του βασικού και του προσεγγιστικού αλγορίθμου

- Επίδραση της παραμέτρου ανοχής απόστασης  $\epsilon$  στο χρόνο εκτέλεσης



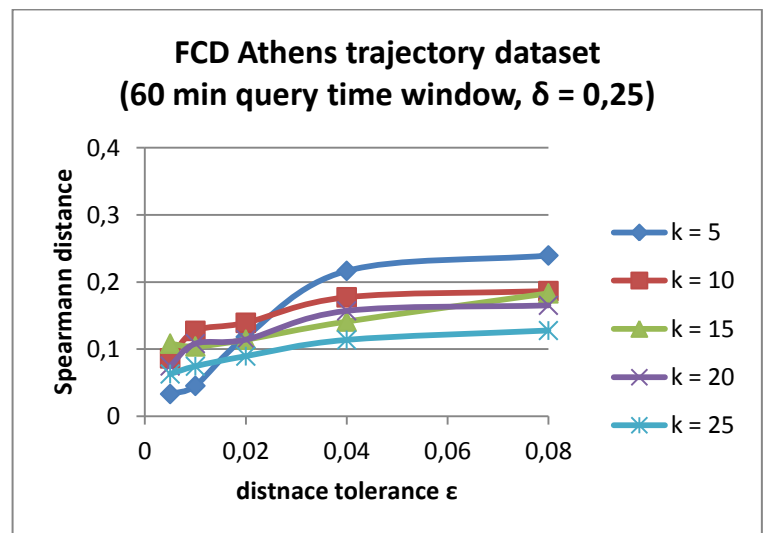
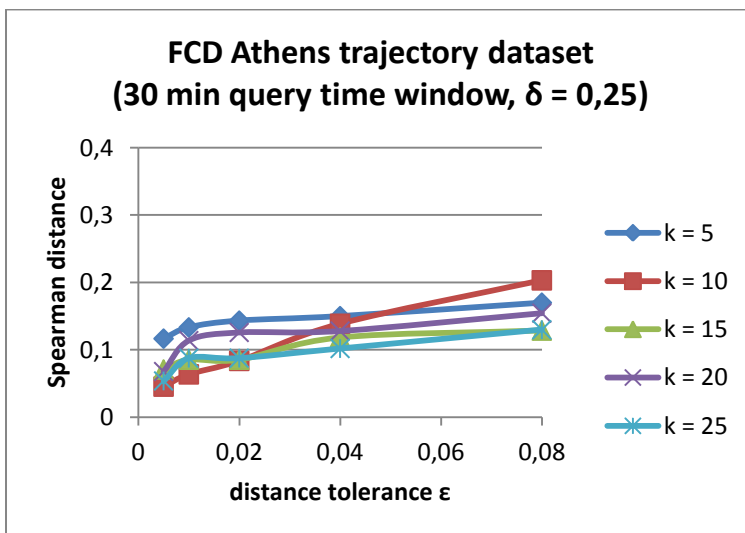
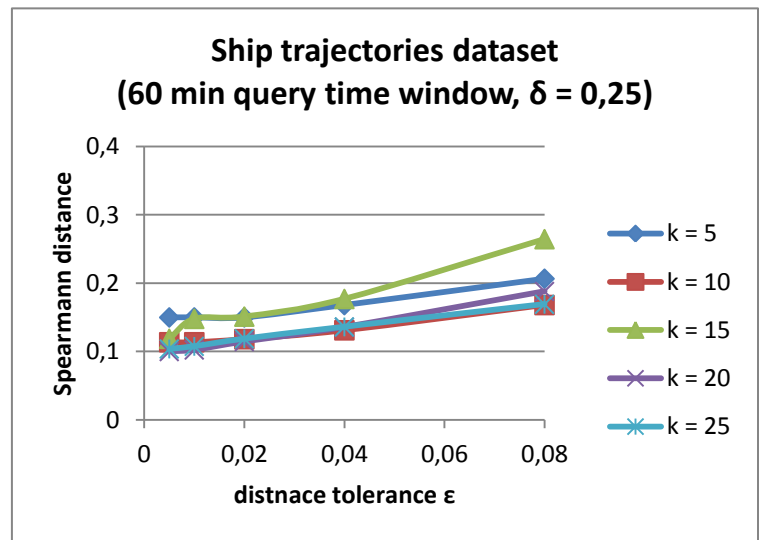
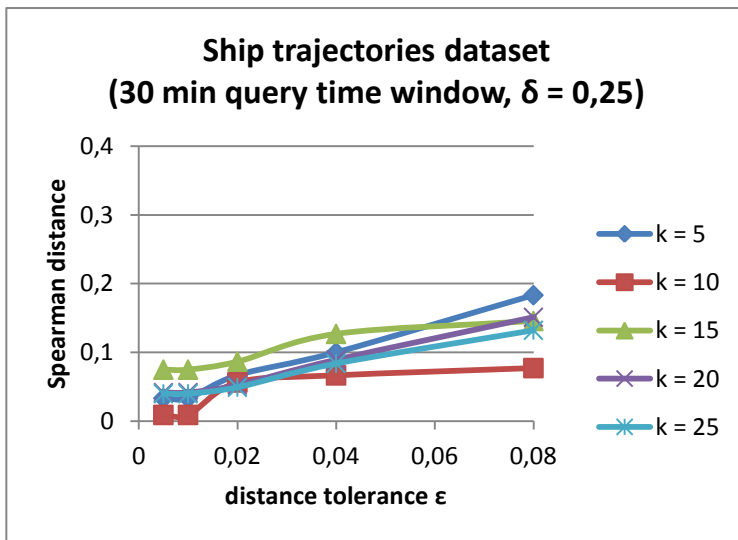
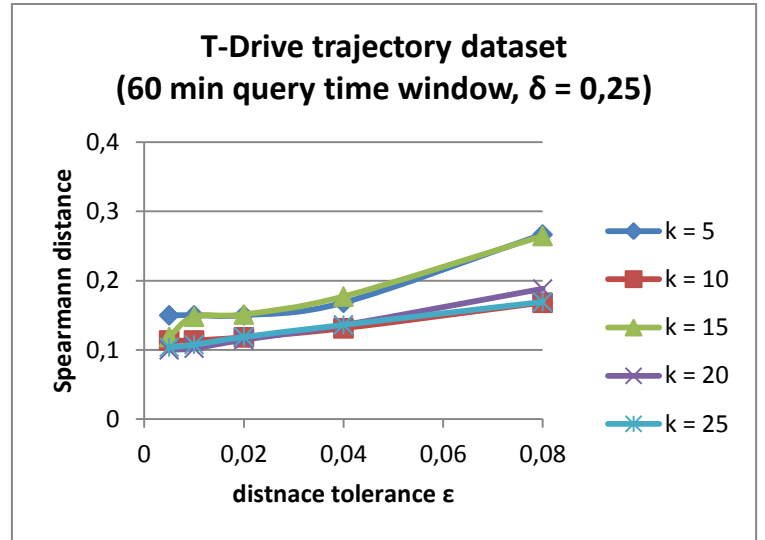
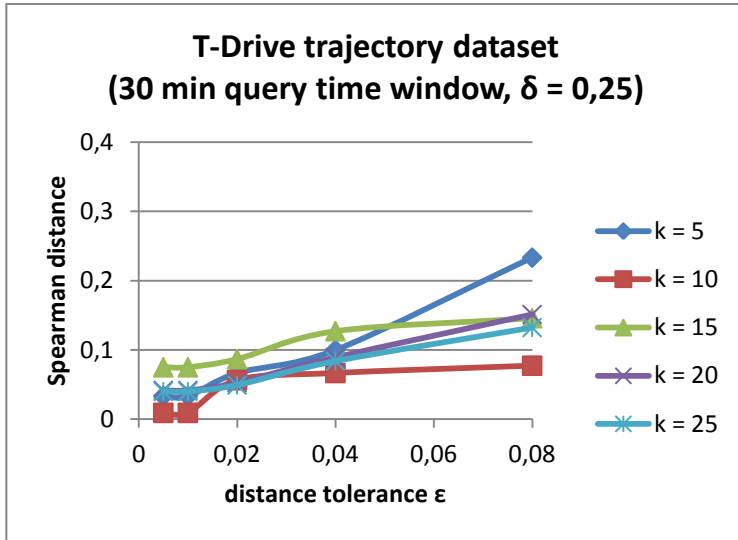
- Επίδραση της παραμέτρου ανοχής ταχύτητας  $\delta$  στο χρόνο εκτέλεσης



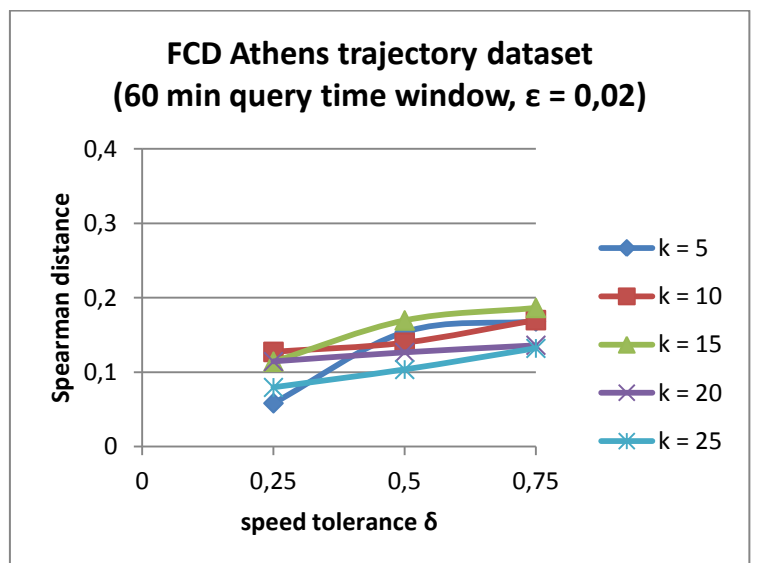
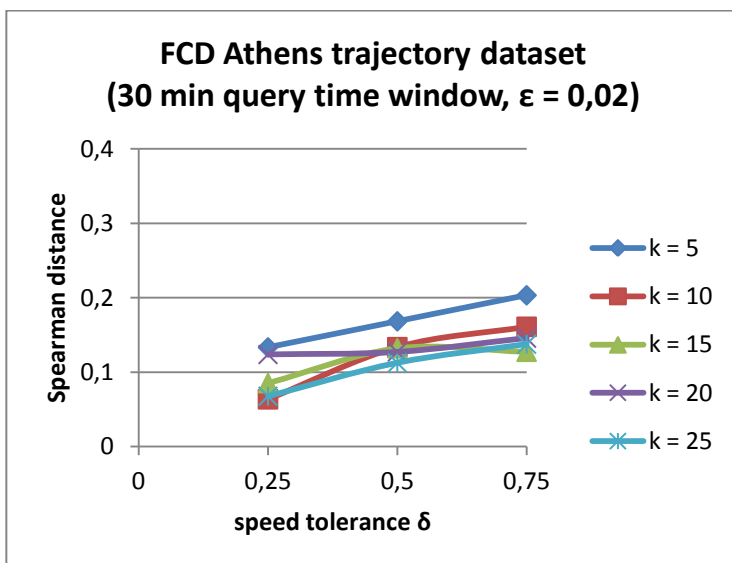
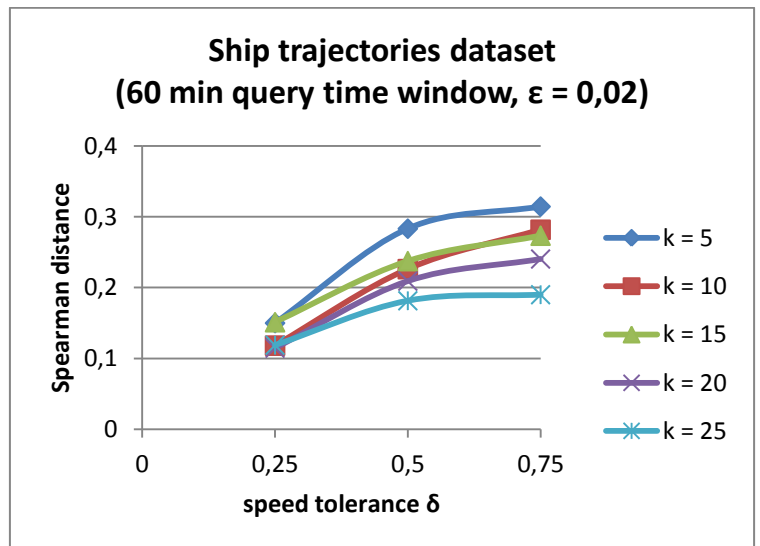
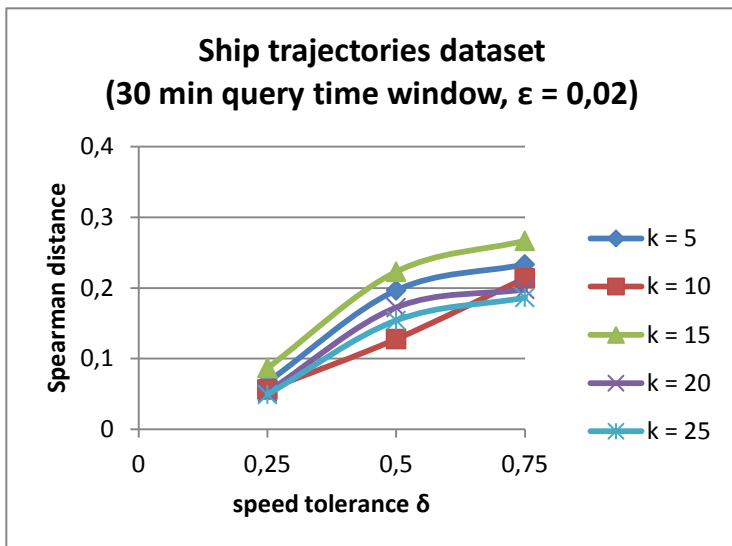
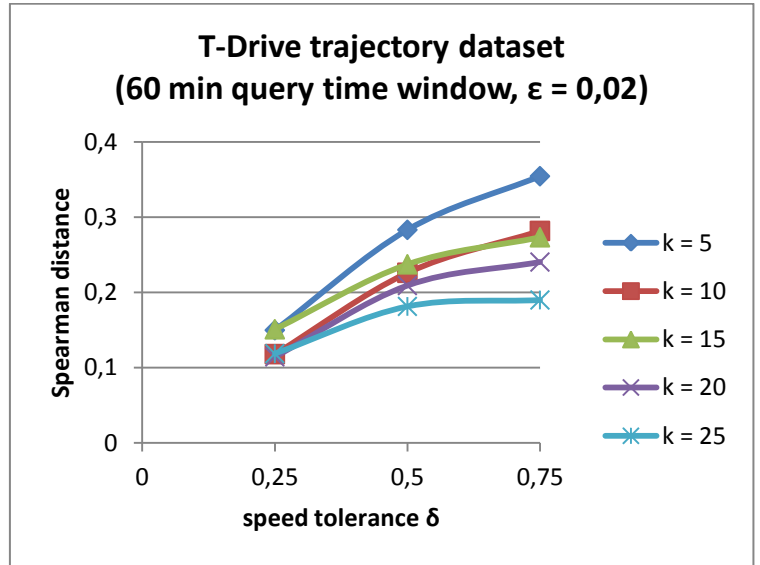
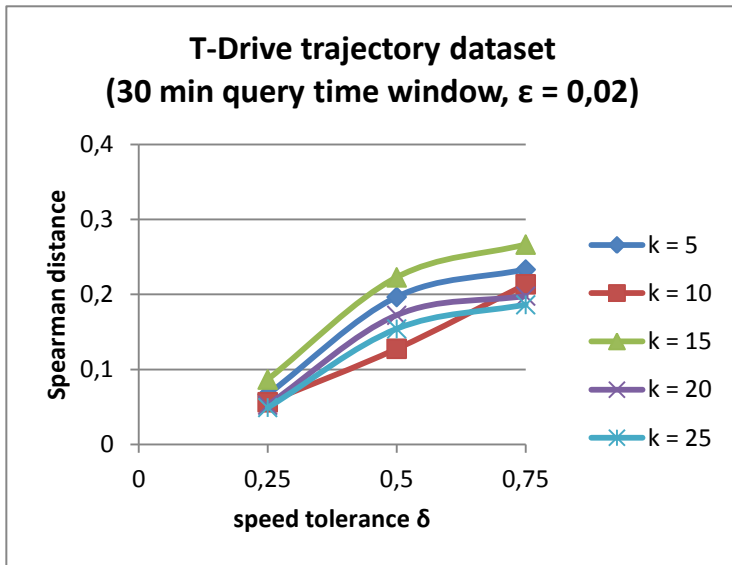
Παρατηρούμε ότι η προσέγγιση των τροχιών μειώνει αρκετά τον αριθμό των τμημάτων τους, με αποτέλεσμα να γίνονται λιγότεροι υπολογισμοί ομοιότητας και να βελτιώνεται ο χρόνος εκτέλεσης. Η τιμή των παραμέτρων ανοχής απόστασης  $\epsilon$  και ταχύτητας  $\delta$  καθορίζει το ποσοστό απλοποίησης της τροχιάς και κατ' επέκταση τη μείωση του χρόνου εκτέλεσης. Ωστόσο, το ποσοστό απλοποίησης διαφέρει για κάθε σύνολο δεδομένων κι εξαρτάται από τη μορφή των τροχιών. Όπως φαίνεται στα διαγράμματα, επιλέγοντας κατάλληλες παραμέτρους ανοχής, ο προσεγγιστικός αλγόριθμος γίνεται ταχύτερος από το βασικό αλγόριθμο με ουρά

προτεραιότητας. Επιπλέον, ο χρόνος εκτέλεσης είναι ανεξάρτητος από τον αριθμό κοντινότερων γειτόνων  $k$ , σε αντίθεση με τον αλγόριθμο με ουρά προτεραιότητας.

▪ Επίδραση της παραμέτρου ανοχής απόστασης  $\epsilon$  στην ακρίβεια



- Επίδραση της παραμέτρου ανοχής ταχύτητας  $\delta$  στην ακρίβεια



Παρατηρούμε ότι αυξάνοντας τις τιμές των παραμέτρων ανοχής απόστασης  $\epsilon$  και ταχύτητας  $\delta$ , αυξάνεται η απόσταση Spearman μεταξύ της προσεγγιστικής και της πραγματικής λίστας κοντινότερων γειτόνων. Συγκεκριμένα, η ανοχή απόστασης  $\epsilon$  μειώνει την ακρίβεια κυρίως της κάθετης και της γωνιακής συνιστώσας απόστασης, ενώ η ανοχή ταχύτητας  $\delta$  επηρεάζει κυρίως την παράλληλη συνιστώσα απόστασης. Επίσης, η ανοχή ταχύτητας  $\delta$  επηρεάζει την ακρίβεια επιλογής ενδιάμεσου σημείου της τροχιάς με γραμμική παρεμβολή σε περίπτωση άνισων τμημάτων.



# 6

## Επίλογος

### 6.1 Σύνοψη και συμπεράσματα

Στην παρούσα εργασία μελετήθηκε το πρόβλημα εύρεσης κ κοντινότερων γειτόνων συγκρίνοντας την ομοιότητα ολόκληρων τμημάτων της τροχιάς αντί για μεμονωμένα σημεία της, όπως συνηθίζεται στη βιβλιογραφία. Για την αποδοτικότερη επίλυση του προβλήματος χρησιμοποιήθηκε ουρά προτεραιότητας ώστε να περιοριστεί ο χώρος αναζήτησης λύσεων. Επίσης, εξετάστηκε η εύρεση προσεγγιστικής λύσης απλοποιώντας τις τροχιές και μειώνοντας έτσι τους υπολογισμούς.

Η πειραματική αξιολόγηση επιβεβαιώνει ότι η συγκεκριμένη προσέγγιση ανάλυσης της χωρο-χρονικής ομοιότητας των τροχιών οδηγεί σε ακριβέστερα αποτελέσματα. Επιπλέον, παρατηρήθηκε ότι η επίλυση του προβλήματος με χρήση ουράς προτεραιότητας μειώνει αισθητά το υπολογιστικό κόστος και ότι ο προσεγγιστικός αλγόριθμος επιτυγχάνει ακόμα ταχύτερα αποτελέσματα με ικανοποιητική ακρίβεια.

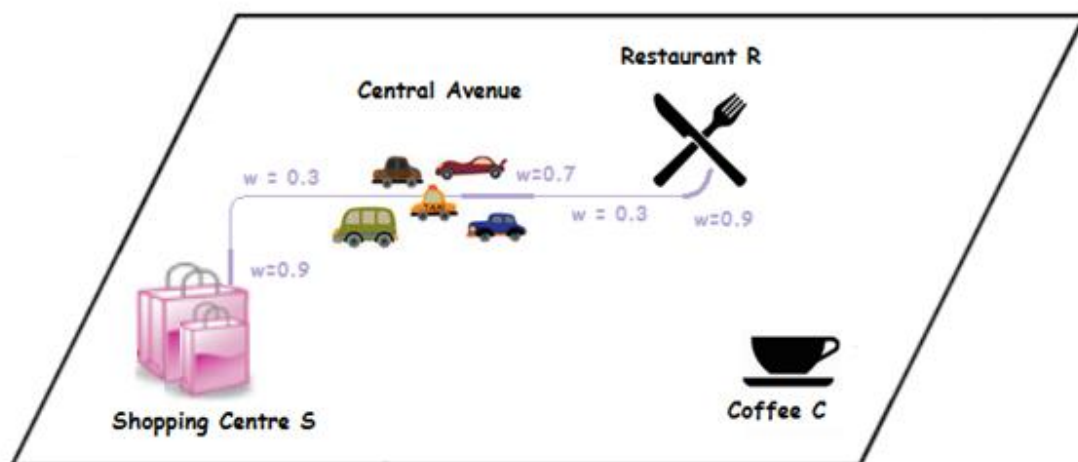
### 6.2 Μελλοντικές επεκτάσεις

#### **Χωρο-χρονική ομοιότητα τροχιών προσανατολισμένη στο χρήστη**

Μια μελλοντική επέκταση της ανάλυσης ομοιότητας τροχιών θα μπορούσε να είναι προσανατολισμένη στις προτιμήσεις του χρήστη και να χρησιμοποιηθεί για να προσφέρει

προσωποποιημένες υπηρεσίες (personalized services) σε μια εφαρμογή. Για παράδειγμα, μια τροχιά μπορεί να περιλαμβάνει τμήματα ιδιαίτερου ενδιαφέροντος που θεωρούνται πιο σημαντικά από άλλα για την ομοιότητα. Σε αυτή την περίπτωση, ο χρήστης θα μπορεί να εισάγει βάρη για τα τμήματα της τροχιάς, τα οποία θα λαμβάνονται υπόψη κατά τους υπολογισμούς ομοιότητας για την εύρεση κοντινότερων γειτόνων σε αυτή.

Η εισαγωγή βαρών από το χρήστη είναι προτιμότερη από τη σημασιολογική σήμανση σημείων ιδιαίτερου ενδιαφέροντος για προσωποποιημένες υπηρεσίες, καθώς προσαρμόζεται στις ιδιαίτερες απαιτήσεις του κάθε χρήστη [WL12]. Αντίθετα, η σημασιολογία βασίζεται σε κοινές απαιτήσεις των περισσότερων χρηστών.



Σχήμα 6.1: Παράδειγμα τροχιάς με βάρη ανάλογα με τις προτιμήσεις του χρήστη

### Δεικτοδότηση (Indexing)

Για τη μείωση του χώρου αναζήτησης λύσεων θα μπορούσε επίσης να χρησιμοποιηθεί κατάλληλη δομή που υποστηρίζει χωρο-χρονική δεικτοδότηση (π.χ grid-based index ή 3D-R-tree). Όμως, η βασική δυσκολία έγκειται στο γεγονός ότι η παραλλαγή της Hausdorff απόστασης που χρησιμοποιείται είναι μη – μετρική συνάρτηση απόστασης.

Μια συνάρτηση απόστασης  $D(I_1, I_2)$  είναι μετρική αν ικανοποιεί τις παρακάτω ιδιότητες [JWG00]:

1.  $D(I_1, I_2) \geq 0$
2.  $D(I_1, I_2) = 0$  αν και μόνο αν  $I_1 = I_2$
3.  $D(I_1, I_2) = D(I_2, I_1)$  (ιδιότητα συμμετρίας)
4.  $D(I_1, I_3) \leq D(I_1, I_2) + D(I_2, I_3)$  (τριγωνική ανισότητα)

Στην ανάλυση ομοιότητας συνηθίζονται να χρησιμοποιούνται μη-μετρικές συναρτήσεις, που είναι πιο αξιόπιστες, ανθεκτικές στο θόρυβο και λαμβάνουν υπόψη την τοπική ομοιότητα.



Για τη δεικτοδότηση μη-μετρικών συναρτήσεων απόστασης απαιτείται κατάλληλη επέκταση των υπαρχουσών δομών, ώστε να εγγυάται η ορθότητα και να μη γίνονται λανθασμένες απορρίψεις (false dismissals) [BS11].



# 7

## Βιβλιογραφία

- [AT10] Aung, H.H., Tan, K.L.: Discovery of evolving convoys. In: Proceedings of the 22nd international conference on Scientific and statistical database management, pp. 196–213 (2010)
- [BS11] Benjamin Bustos, Tomás Skopal: Non-metric similarity search problems in very large collections. ICDE 2011: 1362-1365 (2011)
- [C10] Hakima Chaouchi: The Internet Of Things: Connecting Objects, Wiley-ISTE, May 2010
- [CLR+09] T. Cormen, C. Leiserson, R. Rivest, C. Stein: Introduction to Algorithms, MIT Press, 2009
- [CN04] Chen L., Ng R.: On the marriage of lp-norms and edit distance. In: Proceedings of the international conference on Very large data bases, pp. 792–803 (2004)
- [COO05] Chen L., Ozsu M.T., Oria V.: Robust and fast similarity search for moving object trajectories. In: Proceedings of the ACM SIGMOD international conference on Management of data, pp.491–502 (2005)

- [DP73] Douglas D. , Peucker T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. In *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10 (2), pp. 112–122 (1973)
- [F92] Franklin, C. (1992). An introduction to geographic information systems: Linking maps to databases. *Database*, 15(2) p12
- [FGP+07] Elias Frentzos, Kostas Gratsias, Nikos Pelekis, Yannis Theodoridis: Algorithms for Nearest Neighbor Search on Moving Object Trajectories. *GeoInformatica* 11(2): 159-193 (2007)
- [FKS03] Fagin R., Kumar R., Sivakumar D.: Comparing top k lists. In: *SODA*, pp. 28–36 (2003)
- [GBX10] Ralf Hartmut Güting, Thomas Behr, Jianqiu Xu: Efficient k-nearest neighbor search on moving object trajectories. *VLDB J.* 19(5): 687-714 (2010)
- [GK08] J. Gudmundsson and M. van Kreveld. Computing longest duration flocks in trajectory data. In *Encyclopedia of GIS*, pages 726-732. 2008.
- [JSZ08] H. Jeung, H. T. Shen, and X. Zhou. Convoy queries in spatio-temporal databases. In *ICDE*, pages 1457-1459, 2008.
- [JWG00] David W. Jacobs, Daphna Weinshall, Yoram Gdalyahu: Classification with Nonmetric Distances: Image Retrieval and Class Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22(6): 583-600 (2000)
- [JYZ08] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen. Discovery of convoys in trajectory databases. *PVLDB*, 1(1):1068-1080, 2008.
- [KNB05] P. Kalnis, N. Mamoulis, and S. Bakiras. On discovering moving clusters in spatio-temporal data. In *SSTD*, pages 364-381, 2005.
- [L69] Lang, T. :Rules for robot draughtsmen, *Geographical Magazine*, 42, 50–51 (1969)
- [LCK+00] Seok-Lyong Lee, Seok-Ju Chun, Deok-Hwan Kim, Ju-Hong Lee, Chin-Wan Chung: Similarity Search for Multidimensional Data Sequences. *ICDE 2000*: 599-608

- [LDH+10] Z. Li, B. Ding, J. Han, and R. Kays. Swarm: Mining relaxed temporal moving object clusters. *PVLDB*, 3(1):723-734, 2010.
- [LHW07] Lee, J., Han, J., Whang, K.: Trajectory clustering: a partition-and-group framework. In: *Proceedings of the ACM SIGMOD international conference on Management of data*, pp. 593–604 (2007)
- [LS05] B. Lin, and J. Su: Shapes Based Trajectory Queries for Moving Objects, *Proceedings of ACM GIS*, 2005
- [MB04] N. Meratnia and R. de By. Spatiotemporal Compression Techniques for Moving Point Objects. In *EDBT*, pp. 765-782, March 2004.
- [O81] Opheim, H. ‘Smoothing a digitized curve by data reduction methods’, in *Eurographics ’81*, ed. by Encarnacao, J. L., pp. 127–135, North-Holland Publishing Company, Amsterdam.
- [O90] OGRIP. Advisory Committee's First Year Report. Columbus, OH: Department of Administrative Services, State of Ohio, 1990.
- [PPS06] Michalis Potamias, Kostas Patroumpas, Timos K. Sellis: Sampling Trajectory Streams with Spatiotemporal Criteria. *SSDBM 2006*: 275-284
- [PTM+05] Dimitris Papadias, Yufei Tao, Kyriakos Mouratidis, Chun Kit Hui: Aggregate nearest neighbor queries in spatial databases. *ACM Trans. Database Syst.* 30(2): 529-576 (2005)
- [RH10] Gook-Pil Roh, Seung-won Hwang: NNCluster: An Efficient Clustering Algorithm for Road Network Trajectories. *DASFAA (2) 2010*: 47-61
- [RW74] Reumann K. and Witkam A. P. M.: Optimizing curve segmentation in computer graphics, in *Proceedings of International Computing Symposium*, pp. 467–472, North-Holland Publishing Company, Amsterdam. (1974).
- [SC06] Wenzhong Shi and ChuiKwan Cheung: Performance Evaluation of Line Simplification Algorithms for Vector Generalization, *The Cartographic Journal* Vol. 43 No. 1 pp. 27–44 (2006)

- [SVS10] Lokesh Kumar Sharma, Om Prakash Vyas, Simon Scheider, Ajaya K. Akasapu: Nearest Neighbour Classification for Trajectory Data. ICT 2010: 180-185
- [VBT09] M. R. Vieira, P. Bakalov, and V. J. Tsotras. On-line discovery of flock patterns in spatio-temporal data. In GIS, pages 286-295, 2009.
- [VGK02] Vlachos M., Gunopoulos D., Kollios G.: Discovering similar multidimensional trajectories. In: Proceedings of the IEEE International Conference on Data Engineering, pp. 673–684 (2002)
- [VW93] Visvalingam, M. and Whyatt, J. D. : Line generalization by repeated elimination of points, The Cartographic Journal, 30, 46–51 (1993).
- [WL12] Haibo Wang, Kuien Liu: User oriented trajectory similarity search. Proceedings of the ACM SIGKDD International Workshop on Urban Computing, pages 103-110, 2012
- [WLH06] Y. Wang, E.-P. Lim, and S.-Y. Hwang. Efficient mining of group patterns from user movement data. Data Knowl. Eng., 57(3):240-282, 2006.
- [WFH11] Ian H. Witten, Eibe Frank and Mark A. Hall: Data Mining: Practical Machine Learning Tools and Techniques, Third Edition (2011)
- [YJF98] Yi B.K., Jagadish H.V., Faloutsos C.: Efficient retrieval of similar time sequences under time warping. In: Proceedings of the IEEE International Conference on Data Engineering, pp. 201–208 (1998)
- [YZX+11] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun: Driving with knowledge from the physical world. In The 17th ACM SIGKDD international conference on Knowledge Discovery and Data mining, KDD'11, New York, NY, USA, 2011. ACM.
- [YZZ+10] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang: T-drive: driving directions based on taxi trajectories. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '10, pages 99-108, New York, NY, USA, 2010. ACM.

- [ZS97] Zhao Z., Saalfeld A.: Linear-time sleeve-fitting polyline. In Autocarto 13, ACSM/ASPRS'97 Technical Papers, Seattle, Washington, April 1997, 5, pp. 214–223
- [ZZ11] Zheng Yu, Zhou Xiaofang (Editors): Computing with Spatial Trajectories (2011)