

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ



Ολοκλήρωση χωρικών δεδομένων

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΧΟΥΡΖΑΜΑΝΟΓΛΟΥ ΧΑΡΑΛΑΜΠΟΣ

Επιβλέπων : Χρήστος Δουλκερίδης
Επίκουρος Καθηγητής

Πειραιάς, Οκτώβριος 2019

Η σελίδα αυτή είναι σκόπιμα λευκή.

Περίληψη

Η διαδικασία κατά την οποία, δεδομένα απο διαφορετικές πηγές συνδυάζονται και ανακαλύπτεται κάποια σχέση/σύνδεση μεταξύ τους, μέσω της οποίας προκύπτει νέα χρήσιμη πληροφορία, ονομάζεται ολοκλήρωση δεδομένων.

Ο σκοπός της πτυχιακής εργασίας αυτής είναι η υλοποίηση ενός αλγορίθμου ο οποίος πραγματοποιεί ολοκλήρωση χωρικών δεδομένων. Συγκεκριμένα, έχοντας πηγές ευθύγραμμων τμημάτων, πολυγώνων και σημείων, ο αλγόριθμος πρέπει να ανακαλύπτει το κοντινότερο ευθύγραμμο τμήμα σε κάθε σημείο και ποια πολύγωνα ανήκουν σε κάθε κύκλο με κέντρο κάποιο από τα σημεία και ακτίνα που ορίζεται από εμάς. Κατόπιν, θα γίνει αξιολόγηση της επίδοσης του για διάφορες τιμές παραμέτρων και όγκων δεδομένων και θα παρουσιαστεί επιμέρους σύγκριση, με έναν διαφορετικό βασικό αλγόριθμο, που εξυπηρετεί τον ίδιο σκοπό.

Συγκεκριμένα, θα χρησιμοποιηθούν χωρικά δεδομένα που αναπαριστούν το οδικό δίκτυο όλης της έκτασης της Ελλάδας και δεδομένα που αντιπροσωπεύουν σημεία ενδιαφέροντος και θα εντοπιστούν οι σχέσεις που αναφέρθηκαν παραπάνω, εξάγοντας έτσι νέα πληροφορία.

Abstract

The process in which, data from different sources are combined so that a relation/connection between them is discovered, which in turn will lead to the extraction of new, useful information, is called data integration.

The purpose of this project is the implementation of an algorithm which achieves spatial data integration. Specifically, having vector polygon and point data sources, the algorithm has to discover the nearest vector to every point and the polygons contained in every circle whose center is one of the points and its radius set by us. Then, its performance for different parameter values and data sizes will be evaluated and also, a comparison with a different but basic, spatial data integration algorithm will be presented.

Specifically, spatial data which represent the road network of Greece in its entirety, as well as, data which stand for points of interest, will be used and the aforementioned relations will be determined, thus extracting new information.

Περιεχόμενα

Περιεχόμενα εικόνων	7
Ευχαριστίες	9
Κεφάλαιο 1: Εισαγωγή και στόχος εργασίας.....	10
1.1 Εισαγωγή	10
1.2 Στόχος εργασίας.....	12
Κεφάλαιο 2: Γνωστικό/τεχνολογικό Υπόβαθρο	14
2.1 Χωρικά δεδομένα.....	14
2.2 Χάρτες και χωρικά δεδομένα	18
2.3 Γεωμετρικά αντικείμενα και σχέσεις μεταξύ τους	21
2.3.1 Σχέσεις θέσης	21
2.3.2 Σχέσεις απόστασης	25
2.3.3 Σχέσεις κατεύθυνσης	25
2.3.4 Σχέσεις με βάση την κατηγορία	26
2.3.5 Χώρο-χρονικές σχέσεις.....	26
2.3.6 WKT και γεωμετρικά αντικείμενα	26
2.4 OpenStreetMap και πηγές χωρικών δεδομένων	29
Κεφάλαιο 3: Προσέγγιση προβλήματος και περιγραφή αλγορίθμων	30
3.1 Πλέγμα	30
3.2 Δομές δεδομένων.....	35
3.2.1 Πίνακες	36

3.2.2	Πίνακες κατακερματισμού	37
3.3	Αλγόριθμοι ολοκλήρωσης χωρικών δεδομένων	38
3.3.1	Brute force αλγόριθμος	38
3.3.2	Grid-based αλγόριθμος.....	38
3.4	Πολυπλοκότητα αλγορίθμων	45
Κεφάλαιο 4: Πειραματική μελέτη και υλοποίηση		46
4.1	Comma-separated values (csv)	46
4.2	Πηγές δεδομένων	46
4.3	Java και βιβλιοθήκες	48
4.4	GGIS	50
4.5	Πειραματική μελέτη.....	52
4.5.1	Επίδοση αλγορίθμου για διαφορετικά πλήθη κελιών πλέγματος	52
4.5.2	Σύγκριση επιδόσεων αλγορίθμων grid – brute force	55
Κεφάλαιο 5: Συμπεράσματα και μελλοντικές επεκτάσεις.....		58
Βιβλιογραφία		60

Περιεχόμενα εικόνων

Εικόνα 1: Αναπαράσταση σημείου	16
Εικόνα 2: Αναπαράσταση ευθύγραμμου τμήματος	17
Εικόνα 3: Αναπαράσταση πολυγώνου	17
Εικόνα 4: Αναπαράσταση σημείων, διανυσμάτων και πολυγώνων.....	18
Εικόνα 5: Διαστάσεις	21
Εικόνα 6: Μονοδιάστατη απεικόνιση.....	22
Εικόνα 7: Δισδιάστατη απεικόνιση	22
Εικόνα 8: Disjoint	22
Εικόνα 9: Intersects	23
Εικόνα 10: Touches	23
Εικόνα 11: Within-Contains	24
Εικόνα 12: Equals	24
Εικόνα 13: Crosses	25
Εικόνα 14: MultiPoint	27
Εικόνα 15: MultiLineString	27
Εικόνα 16: MultiPolygon.....	28
Εικόνα 17: GeometryCollection.....	28
Εικόνα 18: Πλέγμα	30
Εικόνα 19: Καθορισμός μεγέθους κελιών με βάση τα βήματα x και y.....	31
Εικόνα 20: Σημείο στο όριο του πλέγματος	32
Εικόνα 21: Κέντρο κύκλου στο όριο του πλέγματος	33

Εικόνα 22: Αναπαράσταση MultiLineString σε πλέγμα	34
Εικόνα 23: Αναπαράσταση εγγεγραμμένου κύκλου και πολυγώνων σε πλέγμα	35
Εικόνα 24: Πίνακες	36
Εικόνα 25: Πίνακες κατακερματισμού.....	37
Εικόνα 26: Γέμισμα πίνακα κατακερματισμού.....	41
Εικόνα 27: Σχέση nearest	43
Εικόνα 28: Σχέση intersects.....	44
Εικόνα 29: Παράδειγμα δομής αρχείου csv	46
Εικόνα 30: Πηγή 1: Οδικό Δίκτυο	47
Εικόνα 31: Πηγή 2: Σημεία.....	47
Εικόνα 32: Πηγή 3: Σημεία Ενδιαφέροντος	48
Εικόνα 33: Πηγή 4: Κύκλοι.....	48
Εικόνα 34: Απεικόνιση της Ελλάδας στο QGIS	50
Εικόνα 35: QGIS μεγέθυνση	51
Εικόνα 36: QGIS id κοντινότερου δρόμου.....	51
Εικόνα 37: QGIS id σημείων ενδιαφέροντος	52
Εικόνα 38: Απόδοση grid αλγορίθμου για τη σχέση nearest.....	53
Εικόνα 39: Απόδοση grid αλγορίθμου για τη σχέση intersects με μεταβλητή ακτίνα.....	54
Εικόνα 40: Απόδοση grid αλγορίθμου για τη σχέση intersects με σταθερή ακτίνα	55
Εικόνα 41: Σύγκριση αλγορίθμων grid-brute force για την σχέση nearest	56
Εικόνα 42: Σύγκριση αλγορίθμων grid-brute force για την σχέση intersects	56
Εικόνα 43: Σύγκριση αλγορίθμων grid-brute force για την σχέση intersects σε line chart	57

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κ. Δουλκερίδη για την συνεχή βοήθεια, που με υπομονή και επιμονή μου προσέφερε καθ'όλη τη διάρκεια υλοποίησης της εργασίας αυτής, χωρίς την οποία η επιτυχής ολοκλήρωση της δεν θα ήταν δυνατή.

Επίσης, θα ήθελα να ευχαριστήσω την οικογένεια μου, για την υποστήριξη που μου παρείχε από την αρχή έως και το πέρας των σπουδών μου.

1

Εισαγωγή και στόχος εργασίας

1.1 Εισαγωγή

Τα τελευταία χρόνια, ο ρόλος των πηγών/βάσεων δεδομένων έχει αλλάξει δραματικά. Από έναν κόσμο, στον οποίο κάθε οργανισμός κατέχει μία κεντρική και σχετικά απομονωμένη βάση για την αποθήκευση και την αρχειοθέτηση δεδομένων, έχουμε μεταβεί σε έναν νέο, άρρηκτα συνδεδεμένο με τον Παγκόσμιο Ιστό (World Wide Web), στον οποίο πολλές διαφορετικές βάσεις δεδομένων, καθώς και άλλες πηγές δομημένης πληροφορίας, πρέπει να αλληλεπιδράσουν και να συνεργαστούν ώστε, ιδανικά, να προσφέρουν στους χρήστες μια πλήρως ολοκληρωμένη σκοπιά. Η διαδικασία κατά την οποία επιτυγχάνονται τα παραπάνω, ονομάζεται ολοκλήρωση δεδομένων.

Η ολοκλήρωση δεδομένων είναι μεγάλης σημασίας, σε έναν κόσμο που, όπως είπαμε, είναι περισσότερο διασυνδεδεμένος από ποτέ, αλλά αναπόφευκτα ετερογενής. Υπάρχουν αμέτρητες πηγές δεδομένων διαθέσιμες σε οργανωμένες βάσεις αλλά και στο διαδίκτυο. Φυσικά, οι πηγές αυτές χρησιμοποιούν διαφορετική γλώσσα και έχουν ποικίλες δομές, αφού έχουν δημιουργηθεί από διαφορετικούς ανθρώπους, σε διαφορετικούς χρόνους και για διαφορετικούς σκοπούς.

Ο σκοπός της ολοκλήρωσης δεδομένων είναι να παρέχει στους χρήστες ενοποιημένη πρόσβαση σε πολλές, ετερογενείς πηγές δεδομένων έτσι ώστε να προσεγγίζεται στον μέγιστο δυνατό βαθμό η εντύπωση μιας μοναδικής πηγής που συνδυάζει τα απαραίτητα στοιχεία όλων και ανταποκρίνεται πλήρως στις ανάγκες του εκάστοτε χρήστη.

Η διεργασία αυτή είναι πλέον αναγκαία σε πληθώρα περιπτώσεων που περιλαμβάνουν εμπορικά αλλά και επιστημονικά πεδία. Η ολοκλήρωση δεδομένων

εμφανίζεται με αυξημένη συχνότητα όσο ο όγκος των δεδομένων μεγαλώνει και η ανάγκη διαμοιρασμού τους γιγαντώνεται.

Μερικά παραδείγματα αποτελούν οι εξής περιπτώσεις: 1) Μία αλυσίδα καταστημάτων θέλει κάθε κατάστημα της να έχει πρόσβαση σε κάθε κίνηση που έγινε σε οποιοδήποτε από τα υπόλοιπα. 2) Κάθε νοσοκομείο του κόσμου να έχει πρόσβαση στο ιστορικό των ασθενών του από όλα τα νοσοκομεία που έχουν επισκευθεί πριν από αυτό. 3) Συνδυασμός ευρημάτων και συμπερασμάτων πολλών ερευνητικών ομάδων ανα τον κόσμο. 4) Ομάδα εταιριών αποφασίζει να συνδυάσει τις βάσεις δεδομένων των επιμέρους εταιριών, ώστε να εξάγει χρήσιμη πληροφορία για την συμπεριφορά των πελατών, με σκοπό να πετύχει καλύτερα οικονομικά αποτελέσματα.

Οι προκλήσεις που είναι αναγκαίο να αντιμετωπιστούν ώστε να επιτευχθεί ολοκλήρωση δεδομένων είναι ποικίλες. Οι κυριότερες από αυτές είναι: 1) Τα αναγκαία δεδομένα κατανέμονται σε διαρκώς αυξανόμενο αριθμό πηγών. 2) Πολλές φορές υπάρχει αναξιοπιστία στα δεδομένα. Κάποια είναι αμφιλεγόμενα, δεν ενημερώνονται με συνέπεια ή είναι ελλιπή. 3) Ο όγκος των δεδομένων αυξάνεται με ταχείς ρυθμούς. 4) Σε ορισμένες περιπτώσεις, όπως live streaming, η ταχύτητα με την οποία δημιουργούνται νέα δεδομένα καθιστά δύσκολη την έγκαιρη ένταξη τους στο ολοκληρωμένο μοντέλο. 5) Τέλος, τα δεδομένα δεν είναι πάντα δομημένα ή έχουν δομές που διαφέρουν σημαντικά μεταξύ τους.[2][3][7]

Στόχος της παρούσας πτυχιακής εργασίας είναι η δημιουργία ενός αποδοτικού αλγορίθμου ο οποίος θα πραγματοποιεί ολοκλήρωση δεδομένων. Ο τύπος δεδομένων που επιλέχθηκαν για το σκοπό αυτό είναι χωρικός (spatial data) που σημαίνει ότι αναπαριστούν την γεωγραφική θέση και τη δομή στοιχείων, όπως για παράδειγμα, σημείων, δρόμων, κτιρίων κλπ.

Η ολοκλήρωση χωρικών δεδομένων έχει πολλές εφαρμογές και μπορεί να προσφέρει χρήσιμη πληροφορία που αφορά θέσεις και διαδρομές οχημάτων, πλοίων και αεροσκαφών, πλοήγηση, εύρεση σημείων ενδιαφέροντος, εντοπισμό παραβιάσεων σε προστατευόμενες ή περιορισμένης πρόσβασης περιοχές κλπ.

Στα πλαίσια της εργασίας αυτής, χρησιμοποιήθηκε μια πηγή δεδομένων που περιέχει ευθύγραμμα τμήματα, μία πηγή που περιέχει πολύγωνα, και δύο πηγές που δημιουργήθηκαν από εμάς, οι οποίες περιέχουν σημεία, είτε μεμονωμένα ή σε συνδυασμό με ακτίνα για την δημιουργία κύκλων.

Συγκεκριμένα, τα ευθύγραμμα τμήματα αντιπροσωπεύουν δρόμους και όλα μαζί απαρτίζουν το οδικό δίκτυο της Ελλάδας, τα πολύγωνα αναπαριστούν σημεία ενδιαφέροντος (points of interest), όπως εστιατόρια, τράπεζες, πρατήρια καυσίμων, σχολεία κλπ και τα σημεία δείχνουν τη γεωγραφική θέση οχημάτων ή ατόμων (στίγμα).

Σκοπός μας είναι, πρώτον, να καθοριστεί, από όλους τους δρόμους του οδικού δικτύου, ποιος είναι ο κοντινότερος σε κάθε όχημα/άτομο, άρα στο γεωγραφικό στίγμα που δείχνει τη θέση του. Δεύτερον, να υπάρχει η δυνατότητα, για οποιοδήποτε σημείο εντός Ελλάδας, ορίζοντας μια ακτίνα, ο αλγόριθμος να επιστρέφει όλα τα σημεία ενδιαφέροντος που βρίσκονται εντός του κύκλου με κέντρο το σημείο και ακτίνα αυτή που έχει οριστεί.

1.2 Στόχος εργασίας

Ο στόχος της παρούσας εργασίας και άρα του αλγορίθμου που αναπτύχθηκε αποτελείται από δύο μέρη.

Πρώτον, επιδιώκεται, για κάθε σημείο της πηγής που περιέχει τα σημεία, να βρεθεί ποιος δρόμος από την πηγή του οδικού δικτύου είναι ο κοντινότερος του. Για να γίνει αυτό, πρέπει το σημείο και ο δρόμος που συγκρίνονται να μετατραπούν σε γεωμετρικά αντικείμενα και να καθοριστεί η μεταξύ τους Ευκλείδεια απόσταση. Η ελάχιστη απόσταση που θα υπολογιστεί μας δείχνει ποιος δρόμος είναι πιο κοντά στο σημείο αυτό και άρα την σχέση απόστασης nearest.

x	y	σχέση nearest
x1	y1	id1
x2	y2	id2
...
x _n	y _n	id _n

Δεύτερον, πρέπει για κάθε κύκλο που περιέχει η τέταρτη πηγή να ανακαλυφθεί ποια σημεία ενδιαφέροντος βρίσκονται εντός του, εξολοκλήρου ή μερικώς. Για να υλοποιηθεί το παραπάνω, πρέπει να μετατραπούν τα σημεία ενδιαφέροντος και οι κύκλοι σε γεωμετρικά αντικείμενα και να βρεθεί, για κάθε κύκλο, ποια πολύγωνα πλοιορούν τις προϋποθέσεις της σχέσης θέσης intersects, μαζί του.

x	y	ακτίνα	σχέση intersects
x1	y1	r1	id1,id2,...,id _k
x2	y2	r2	id1,id2,...,id _m
...
x _n	y _n	r _n	id1,id2,...,id _p

Παρόλο που η παραπάνω προσέγγιση λειτουργεί, έχει ένα βασικό μειονέκτημα. Για να υπολογιστούν τα απαιτούμενα μεγέθη και να ανακαλυφθούν οι εν λόγω σχέσεις, είναι αναγκαίο κάθε σημείο να συγκριθεί με κάθε δρόμο και κάθε κύκλος να συγκριθεί με κάθε σημείο ενδιαφέροντος που εμπεριέχεται στα δεδομένα μας. Το αλγοριθμικό κόστος αυτού είναι πολύ μεγάλο και ειδικά εάν επιθυμητό να επαναληφθεί η διαδικασία για μεγάλο όγκο σημείων και κύκλων.

Για να ξεπεραστεί το εμπόδιο αυτό, έπρεπε να βρεθεί ένας τρόπος ο οποίος να μας επιτρέπει, για κάθε σημείο και για κάθε κύκλο, να κάνουμε τους προαναφερθέντες ελέγχους, όχι για όλους τους δρόμους και τα σημεία ενδιαφέροντος αντίστοιχα, αλλά μόνο για αυτά που βρίσκονται κοντά, εντός μιας εμβέλειας, σε κάθε σημείο και κύκλο. Έτσι, μειώνεται δραματικά το κόστος και άρα ο χρόνος εκτέλεσης του αλγορίθμου.

Ο τρόπος επίτευξης των παραπάνω και συνεπώς η λειτουργία του αλγορίθμου, θα αναπτυχθεί σε επόμενο κεφάλαιο.

2

Γνωστικό/Τεχνολογικό υπόβαθρο

Στο κεφάλαιο αυτό θα αναλυθούν τα χωρικά δεδομένα και η σχέση τους με τους χάρτες, θα εξεταστούν τα είδη γεωμετρικών αντικειμένων που προκύπτουν από αυτά και πως σχετίζονται μεταξύ τους και θα γίνει αναφορά στο OpenStreetMap.

2.1 Χωρικά δεδομένα

Ανέκαθεν, υπήρχε η ανάγκη να γνωρίζουμε τη θέση αντικειμένων στο χώρο. Που βρίσκονται σε σχέση με μας αλλά και σε σχέση με άλλα αντικείμενα, δηλαδή την τοποθεσία τους. Χωρίς την γνώση αυτή, η πλοήγηση μέσα σε μία πόλη, μεταξύ πόλεων, μεταξύ χωρών κ.ο.κ θα ήταν αδύνατη.

Τα δεδομένα που σχετίζονται με την τοποθεσία ονομάζονται χωρικά (spatial) ή γεωγραφικά. Η γεωγραφική πληροφορία, διαφέρει από άλλου είδους πληροφορίες στο ότι τα δεδομένα αναφέρονται σε αντικείμενα ή φαινόμενα με συγκεκριμένη θέση στο χώρο, που έχουν δηλαδή χωρική διεύθυνση. Λόγω αυτού του ιδιαίτερου χαρακτηριστικού, οι τοποθεσίες των αντικειμένων ή φαινομένων μπορούν να οπτικοποιηθούν, και οι οπτικοποιήσεις αυτές, που ονομάζονται χάρτες, είναι το κλειδί για την περαιτέρω ανάλυση τους.

Τα γεωχωρικά δεδομένα χωρίζονται σε τρεις κατηγορίες. Η πρώτη (locational data), αφορά τα γεωμετρικά χαρακτηριστικά (θέση και διαστάσεις) του εν λόγω αντικειμένου. Η δεύτερη (attribute data), αναφέρεται σε άλλα, μη γεωμετρικά χαρακτηριστικά του. Η τρίτη (temporal data), αναφέρεται στη χρονική στιγμή ή χρονικό διάστημα που οι δύο προηγούμενες κατηγορίες δεδομένων είναι έγκυρες. Στα πλαίσια της εργασίας αυτής, τα χωρικά δεδομένα που θα χρησιμοποιηθούν και θα αναλυθούν ανήκουν στην πρώτη κατηγορία.

Τα δεδομένα αυτά προσδιορίζουν μοναδικά τη γεωγραφική θέση αλλά και τη μορφή στοιχείων πάνω στη Γη, χρησιμοποιώντας κάποιο σύστημα συντεταγμένων. Τις γεωγραφικές συντεταγμένες αποτελούν το γεωγραφικό πλάτος (latitude) και το γεωγραφικό μήκος (longitude). Γεωγραφικό πλάτος ενός σημείου στην επιφάνεια της γης είναι η γωνία που σχηματίζεται από το επίπεδο του ισημερινού και την κατακόρυφο του τόπου. Γεωγραφικό μήκος ενός σημείου στην επιφάνεια της γης είναι η γωνία που σχηματίζεται από το επίπεδο του μεσημβρινού που διέρχεται από το εν λόγω σημείο με το επίπεδο του πρώτου μεσημβρινού. Γεωγραφικό στίγμα (geographical position) ενός σημείου είναι η τομή των παραπάνω γεωγραφικών συντεταγμένων.[1]

Για τη δημιουργία, επεξεργασία, αποθήκευση και οπτικοποίηση των χωρικών δεδομένων η έχουν αναπτυχθεί πληροφοριακά συστήματα που ονομάζονται Γεωγραφικά Πληροφοριακά Συστήματα (Geographical Information Systems – GIS). Τα συστήματα αυτά προσφέρουν πολύτιμη βοήθεια σε ποικίλους τομείς όπως ναυσιπλοΐα, δασοκομία, διαχείριση καταστροφών, φυσικούς πόρους, υγεία κλπ. Οι πληροφορίες που συλλέγονται από τη χρήση τους μοιράζονται μεταξύ χρηστών, εφαρμογών και άλλων συστημάτων. Όσο η λειτουργικότητα τους συνέχιζε να ωριμάζει, η χρήση τους επεκτάθηκε σε όλους τους τομείς που ασχολούνται με γεωχωρικά δεδομένα. Τα GIS εισήγαγαν την ολοκλήρωση χωρικών δεδομένων από διάφορες πηγές και άρα έδωσαν την δυνατότητα στους χρήστες να δώσουν απαντήσεις σε σύνθετα ερωτήματα βασιζόμενοι στα συνδυασμένα δεδομένα.

Οι τύποι απαντήσεων που μπορούν να δώσουν τα GIS είναι οι εξής:

- *Αναγνώριση:* Επιλέγοντας μια θέση στο χώρο, επιστρέφεται ένα όνομα ή κάποια άλλη πληροφορία που χαρακτηρίζει οτιδήποτε υπάρχει εκεί.
- *Τοποθεσία:* Ανάλογα τα κριτήρια της ερώτησης η απάντηση μπορεί να είναι συντεταγμένες ενός ή περισσότερων αντικειμένων.
- *Τάσεις:* Τι έχει αλλάξει από μια χρονική στιγμή μέχρι κάποια άλλη. Για παράδειγμα η δομική ανάπτυξη μιας περιοχής σε διάστημα μερικών δεκαετιών.
- *Βέλτιστο μονοπάτι:* Ποιός είναι ο πιο σύντομος ή οικονομικός δρόμος για να μεταβεί κάποιος από ένα σημείο σε ένα άλλο.

- *Μοτίβα*: Τι σχέσεις υπάρχουν μεταξύ αντικειμένων ή κλάσεις αντικειμένων. Για παράδειγμα, η σχέση μεταξύ του αριθμού εργοστασίων και των χαρακτηριστικών των κτιρίων σε μια περιοχή.
- *Μοντέλα*: Δημιουργούνται με βάση την υπάρχουσα γνώση και χρησιμοποιούνται με σκοπό την πρόβλεψη μελλοντικών συνθηκών.

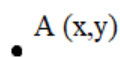
Παρόλη τη σημαντική μείωση του κόστους του λογισμικού αλλά και του hardware με την πάροδο του χρόνου, τα συλλεγόμενα χωρικά δεδομένα αυξάνονται με ταχείς ρυθμούς και η συνεχής ανάπτυξη και συντήρηση των απαραίτητων υποδομών είναι ιδιαίτερα απαιτητική.[1][11][9]

Τα συστήματα αυτά υποστηρίζουν διάφορες μορφές χωρικών δεδομένων και πολλούς τύπους αρχείων. Στα GIS τα δεδομένα χωρίζονται σε δύο μεγάλες κατηγορίες: στα διανυσματικά (vector) δεδομένα και στα ψηφιδωτά (raster) δεδομένα.

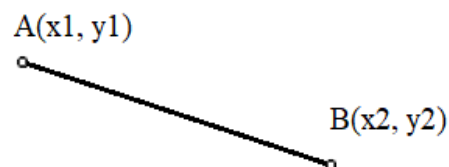
Τα διανυσματικά δεδομένα αναπαριστούν τριών ειδών χωρικές οντότητες με βάση το σχήμα τους:

- Σημεία (Points)
- Ευθύγραμμα τμήματα/ διανύσματα (Polylines)
- Πολύγωνα (Polygons)

Τα σημεία αποτελούν την βασική χωρική οντότητα με την βοήθεια της οποίας αναπαρίστανται οι επόμενες. Έχουν συγκεκριμένη θέση στο χώρο (στίγμα) και μπορούν να αναπαραστήσουν είτε ένα φυσικό αντικείμενο (όπως ένα όχημα ή ένα μνημείο) ή κάποιο συμβάν (π.χ. τροχαίο ατύχημα).



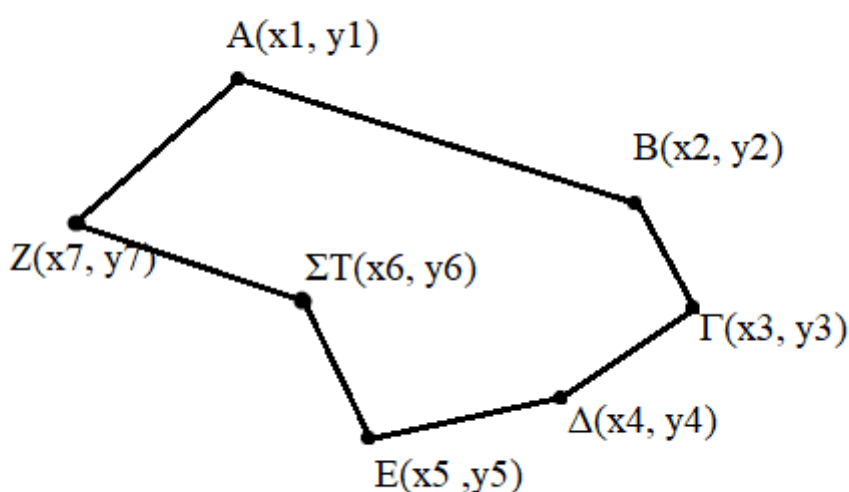
Εικόνα 1- Αναπαράσταση σημείου



Εικόνα 2- Αναπαράσταση ευθύγραμμου τμήματος

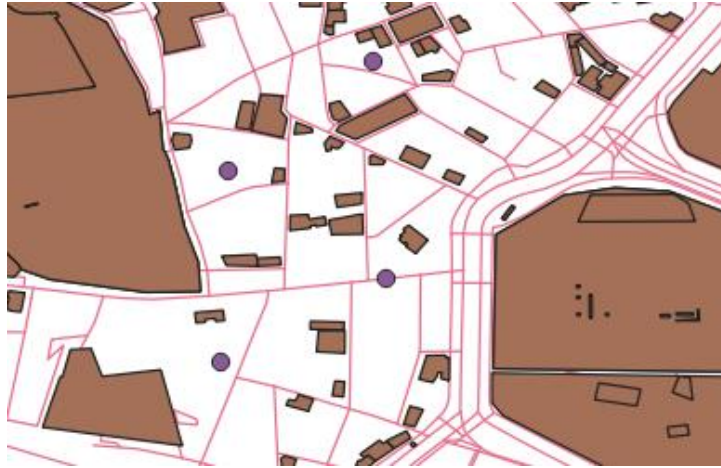
Τα ευθύγραμμα τμήματα ορίζονται από τουλάχιστον δύο σημεία και μπορούν, όπως και τα σημεία, να αναπαραστήσουν κάποιο φυσικό αντικείμενο (όπως δρόμους ή καλώδια) ή μια κίνηση όπως για παράδειγμα, η διαδρομή ενός οχήματος κατά μήκος του οδικού δικτύου.

Τα πολύγωνα, είναι χωρικές οντότητες, οι οποίες αποτελούνται από πολλά ευθύγραμμα τμήματα/διανύσματα και άρα πολλά σημεία, και η αρχή του πρώτου διανύσματος ταυτίζεται με το τέλος του τελευταίου διανύσματος. Τα πολύγωνα είναι κλειστά γεωμετρικά σχήματα που αντιπροσωπεύουν φυσικά αντικείμενα μεγάλης επιφάνειας που δεν είναι δόκιμο να αναπαρασταθούν με ένα σημείο, όπως για παράδειγμα κτίρια ή συμπλέγματα κτιρίων, γήπεδα, οικόπεδα, πάρκα, άλση ή σε μεγαλύτερη κλίμακα, λίμνες, βουνά, νομοί ή χώρες.



Εικόνα 3- Αναπαράσταση πολυγώνου

Ακολουθεί μία οπτική αναπαράσταση και των τριών τύπων διανυσματικών χωρικών δεδομένων.



Εικόνα 4- Αναπαράσταση σημείων, διανυσμάτων και πολυγώνων

Τα ψηφιδωτά δεδομένα αναπαριστούν επιφάνειες. Η αναπαράσταση αυτή βασίζεται σε ψηφίδες όπου η καθεμία αποτελεί ένα τετράγωνο με συγκεκριμένες διαστάσεις που αντιπροσωπεύει ένα τμήμα της επιφάνειας της γης. Τα πιο διαδεδομένα ψηφιδωτά δεδομένα είναι οι δορυφορικές εικόνες και οι αεροφωτογραφίες όπου η κάθε ψηφίδα μπορεί να έχει διαστάσεις από μερικά εκατοστά ως μερικές δεκάδες μέτρα και μια τιμή που αφορά στο χρώμα που φωτογράφησε ο δορυφόρος ή το αεροπλάνο για ένα μικρό τμήμα της επιφάνειας της γης.

Στην παρούσα εργασία, θα ασχοληθούμε αποκλειστικά με διανυσματικά χωρικά δεδομένα.

2.2 Χάρτες και χωρικά δεδομένα

Οι χάρτες χρησιμοποιούνται για να οπτικοποιήσουμε χωρικά δεδομένα, δεδομένα δηλαδή που αναφέρονται στην τοποθεσία ή τα χαρακτηριστικά αντικειμένων ή φαινομένων πάνω στη Γη. Βοηθούν τους χρήστες τους να κατανοήσουν καλύτερα τις υπάρχουσες γεωχωρικές σχέσεις. Από τους χάρτες μπορούν να εξαχθούν πληροφορίες για αποστάσεις, κατευθύνσεις και μεγέθη περιοχών, να αποκαλυφθούν μοτίβα και να κατανοηθούν και να ποσοτικοποιηθούν διάφορων ειδών σχέσεις.

Οι χάρτες δεν είναι πλέον το τελικό προϊόν που ήταν κάποτε. Αρχικά, λειτουργούσαν ως ένα μέσο, πρώτον, για αποθήκευση και δεύτερον, για παρουσίαση γεωχωρικών δεδομένων. Η εισαγωγή του ηλεκτρονικού υπολογιστή και άρα των χαρτών ψηφιακής μορφής και των αντίστοιχων βάσεων δεδομένων τους, διαχώρισε τις δύο αυτές λειτουργίες. Επίσης, η εξέλιξη της τεχνολογίας των βάσεων δεδομένων και των

γραφικών έδωσε στους δημιουργούς χαρτών την δυνατότητα για νέες, εναλλακτικές μορφές παρουσίασης, όπως είναι οι τρισδιάστατοι και διαδραστικοί χάρτες. Το διαδίκτυο έφερε την επόμενη επανάσταση στο χώρο. Πλέον η πρόσβαση σε διαδραστικούς ψηφιακούς χάρτες και ποικίλα εργαλεία για την εύρεση χρήσιμων πληροφοριών δεν περιορίζεται στους επαγγελματίες.

Η επιστήμη που ασχολείται με την κατασκευή και μελέτη χαρτών ονομάζεται Χαρτογραφία. Θα ακολουθήσει μια σύντομη αναφορά στα χαρακτηριστικά εκείνα των χαρτών που σχετίζονται με τα χωρικά δεδομένα: i) την κλίμακα, ii) την προβολή και iii) το σύστημα συντεταγμένων.

Ένας χάρτης συνήθως αναπαριστά μια περιοχή της γης σε χαρτί ή στην οθόνη του υπολογιστή και άρα είναι πολύ μικρότερος από το πραγματικό της μέγεθος. Για να έχουμε τη δυνατότητα όμως, να κατανοούμε ποιό είναι το πραγματικό μέγεθος της περιοχής που αναπαρίσταται και να υπολογίζουμε πραγματικές αποστάσεις με βάση το χάρτη, είναι απαραίτητο κάθε χάρτης να περιλαμβάνει την αναλογία πραγματικής απόστασης στην οποία αντιστοιχεί μια μονάδα μέτρησης του. Η αναλογία αυτή ονομάζεται κλίμακα (scale) του χάρτη.

Δεδομένου ότι η κλίμακα είναι το κλάσμα χάρτη προς την πραγματικότητα, όσο αυτό το κλάσμα μικραίνει, τόσο μειώνεται και η λεπτομέρεια και αντίστροφα. Για παράδειγμα, σε ένα χάρτη μεγάλης κλίμακας μια πόλη μπορεί να αναπαρασταθεί από μια σειρά πολυγώνων που καθένα αφορά σε ένα οικοδομικό τετράγωνο, ενώ σε ένα χάρτη μικρής κλίμακας μια πόλη μπορεί να αναπαρασταθεί ως ένα σημείο που έχει τις συντεταγμένες του κέντρου της πόλης αυτής.[10]

Χαρτογραφική προβολή (projection), είναι ο γεωμετρικός μετασχηματισμός κατά τον οποίο η σφαιρική επιφάνεια της Γης προβάλλεται στο επίπεδο ώστε να είναι δυνατή η εργασία στον υπολογιστή ή με κάποιο φυσικό αντίγραφο του χάρτη, τυπωμένο σε χαρτί.

Τα Συστήματα Συντεταγμένων (Coordinate Systems) έχουν κατασκευαστεί έτσι ώστε να πλαισιώνονται από αυστηρούς κανόνες, με βάση τους οποίους να μπορούν να προσδιοριστούν θέσεις και αποστάσεις και κατά συνέπεια η θέση ενός σημείου στο χώρο καθώς και η απόσταση δυο σημείων στο χώρο. Τα βασικά είδη συστημάτων συντεταγμένων είναι δύο:

- Σύστημα Γεωγραφικών Συντεταγμένων
- Σύστημα Ορθογώνιων / Επιπέδων Συντεταγμένων

Στο Σύστημα Γεωγραφικών Συντεταγμένων (Geographical Coordinate System) ένα σημείο προσδιορίζεται από το γεωγραφικό πλάτος και μήκος, όπως αναφέραμε παραπάνω. Το σύστημα αυτό έχει οριστεί από τους αρχαίους Έλληνες γεωγράφους-φιλοσόφους και οι συντεταγμένες είναι γωνίες με μονάδες μέτρησης τις μοίρες, λεπτά, δευτερόλεπτα και τις υποδιαίρέσεις τους. Χρησιμοποιείται ευρέως στη ναυσιπλοΐα, την τοπογραφία και σε εφαρμογές όπως το Google Earth.

Το Σύστημα Ορθογώνιων Συντεταγμένων ή Καρτεσιανό Σύστημα Συντεταγμένων (Cartesian coordinate system) χρησιμοποιείται σε επίπεδες χαρτογραφικές αναπαραστάσεις της σφαιρικής Γης. Συνήθως αφορά ένα τμήμα της (π.χ. μια χώρα) και βοηθά στην εύρεση θέσεων και αποστάσεων σε ορθογώνιο σύστημα αξόνων δύο ή τριών διαστάσεων. Το σύστημα αυτό παρουσιάστηκε πρώτη φορά στην κινεζική χαρτογραφία τον 3ο αιώνα μΧ. και είναι ευρέως διαδεδομένο λόγω του ότι οι καρτεσιανές συντεταγμένες μπορούν να χρησιμοποιηθούν από την Ευκλείδεια γεωμετρία για τον υπολογισμό αποστάσεων με βάση το πυθαγόρειο θεώρημα. Είναι μια μαθηματική κατασκευή που ορίζεται από μια αρχή 0 και μια μονάδα μέτρησης. Εμπεριέχει ένα σύστημα κάθετων αξόνων X, Y με σημείο τομής τους το 0. Κάθε υποδιαίρεση του άξονα αποτελεί μονάδα απόστασης από την αρχή του. Αν τοποθετήσουμε ένα σημείο K στο σύστημα αυτό, μπορούμε να βρούμε τη θέση του με τον εξής τρόπο: η οριζόντια απόσταση από τον άξονα των Y ονομάζεται τετμημένη (x) και η κάθετη απόσταση από τον άξονα των X ονομάζεται τεταγμένη (y). Επομένως οι συντεταγμένες του σημείου K είναι (x,y) και αντιστοιχούν στο γεωγραφικό μήκος και πλάτος του σημείου στη σφαιρική Γη που όμως πλέον έχει προβληθεί στο επίπεδο. Αν τοποθετήσουμε δύο σημεία A και B στο σύστημα αυτό, με συντεταγμένες (x1, y1) και (x2, y2) αντίστοιχα μπορούμε να υπολογίσουμε την απόστασή τους με βάση το πυθαγόρειο θεώρημα. Η απόσταση αυτή ονομάζεται Ευκλείδεια και δίνεται από τον μαθηματικό τύπο:

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Ανάλογα με το σύστημα προβολής και την επιθυμητή ακρίβεια ορίζεται και το σύστημα συντεταγμένων. Έτσι, κάθε περιοχή του πλανήτη (ήπειρος, χώρα) έχει το δικό της σύστημα προβολής και σύστημα συντεταγμένων.

Στην εργασία αυτή, χρησιμοποιήθηκε το WGS84 (World Geodetic System 1984) ως βάση για τον γεωμετρικό μετασχηματισμό μέσω του οποίου έγιναν οι απαραίτητες μετατροπές στις γεωγραφικές συντεταγμένες, ώστε να μπορούν να υπολογιστούν σωστά τα μεγέθη που χρειάζονταν.

2.3 Γεωμετρικά αντικείμενα και σχέσεις μεταξύ τους

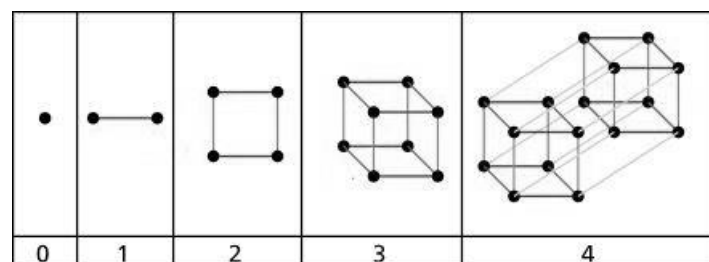
Όλοι μας είμαστε εξοικειωμένοι με βασικά και μη γεωμετρικά αντικείμενα, ο κόσμος γύρω μας άλλωστε είναι γεμάτος με αυτά. Μπορούν να αναπαρασταθούν από τους τύπους διανυσματικών χωρικών δεδομένων που αναφέραμε σε προηγούμενη ενότητα (σημείο, ευθύγραμμο τμήμα, πολύγωνο). Από εδώ και στο εξής θα αναφερόμαστε σε ένα γεωμετρικό αντικείμενο απλά ως γεωμετρία.

Ορισμένες φορές είναι επιθυμητό να γνωρίζουμε ποια είναι η σχέση που συνδέει δύο γεωμετρίες.

2.3.1 Σχέσεις θέσης

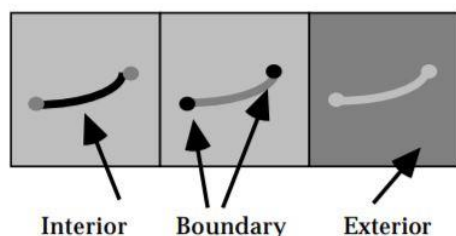
Οι σχέσεις θέσης ή τοπολογικές σχέσεις, είναι αυτές που προκύπτουν εξετάζοντας πως συνδέονται οι θέσεις που κατέχουν δύο γεωμετρίες στο χώρο. Πριν τον ορισμό των σχέσεων, θα γίνει μια αναφορά στις διαστάσεις και τις επιμέρους περιοχές των χωρικών/γεωμετρικών αντικειμένων.

Ως διάσταση ενός αντικειμένου στο χώρο ορίζεται ο ελάχιστος αριθμός συντεταγμένων που είναι αναγκαίες για να περιγραφεί ένα οποιοδήποτε σημείο του. Οι διαστάσεις στον χώρο απεικονίζονται παρακάτω.

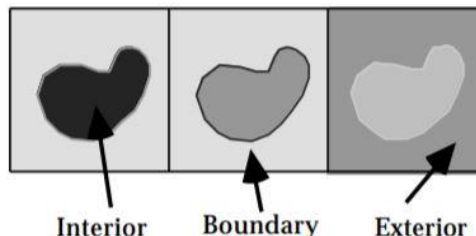


Εικόνα 5 – Διαστάσεις

Οι περιοχές που απαρτίζουν ένα χωρικό αντικείμενο είναι το εσωτερικό του μέρος (interior), δηλαδή η περιοχή εντός του ορίου του, το εξωτερικό (exterior), δηλαδή η περιοχή εκτός του ορίου του και το ίδιο το όριο του (boundary), το οποίο διαχωρίζει το εσωτερικό από το εξωτερικό. Παρακάτω απεικονίζονται οι εν λόγω περιοχές για μονοδιάστατα και δισδιάστατα χωρικά αντικείμενα αντίστοιχα.



Εικόνα 6 – Απεικόνιση μίας διάστασης



Εικόνα 7 – Επικόνιση δύο διαστάσεων

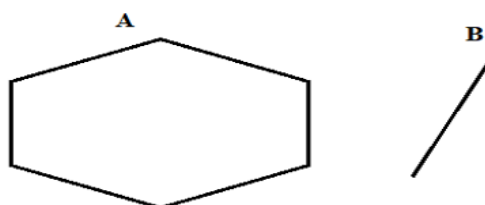
Για κάθε δύο γεωμετρικά αντικείμενα a, b που μπορεί να είναι σημεία, ευθύγραμμα τμήματα ή πολύγωνα, οι βασικότερες σχέσεις είναι οι εξής:

Disjoint

Οι γεωμετρίες A, B δεν έχουν κοινά κανένα κοινό σημείο και άρα δεν τέμνονται με οποιονδήποτε τρόπο. Σχηματίζουν ένα σύνολο αποσυνδεδεμένων γεωμετριών.

$$a \cap b = \emptyset$$

Αν a, b σύνολα που αποτελούνται από τα σημεία που απαρτίζουν τις A και B αντίστοιχα, τότε η τομή των a και b είναι το κενό σύνολο.



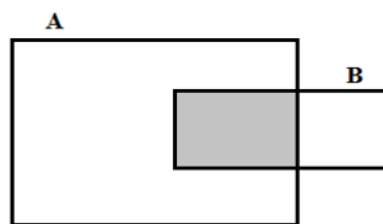
Εικόνα 8- Disjoint

Intersects

Οι γεωμετρίες έχουν τουλάχιστον ένα κοινό σημείο. Μπορεί να έχουν ένα, κάποια, ή όλα τα σημεία τους κοινά. Αξίζει να αναφερθεί ότι για να ισχύει οποιαδήποτε τοπολογική σχέση, εκτός της disjoint, πρέπει να ισχύει η intersects.

$$a \cap b \neq \emptyset$$

Εδώ η τομή των συνόλων a,b δεν είναι το κενό σύνολο.



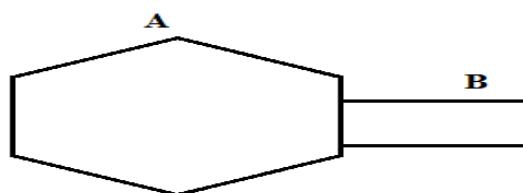
Εικόνα 9- Intersects

Touches

Οι γεωμετρίες απλώς αγγίζουν η μία την άλλη. Υπάρχει μόνο επαφή των ορίων τους και η μία δεν εισέρχεται στην άλλη. Έχουν τουλάχιστον ένα οριακό σημείο κοινό, αλλά κανένα εσωτερικό.

$$(a \cap b \neq \emptyset) \wedge (a^\circ \cap b^\circ = \emptyset)$$

Ως a°, b° ορίζονται τα σύνολα που αποτελούνται από όλα τα εσωτερικά σημεία των γεωμετριών A,B αντίστοιχα. Άρα η τομή των συνόλων των σημείων των A,B δεν είναι το κενό σύνολο, άρα έχουν κοινά σημεία και ταυτόχρονα η τομή των συνόλων των εσωτερικών σημείων τους είναι το κενό. Άρα τα όρια τους, και μόνο αυτά, έχουν κοινά σημεία.



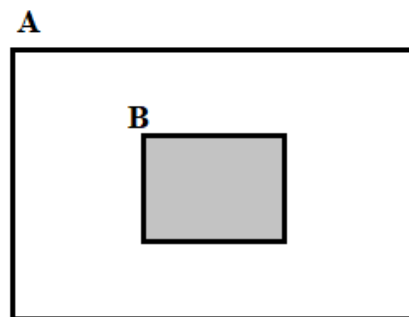
Εικόνα 10- Touches

Within – Contains

Η μία γεωμετρία βρίσκεται εντός της άλλης. Τα κοινά τους σημεία των δύο γεωμετριών είναι όλα τα σημεία που απαρτίζουν την εσωτερική. Εδώ βλέπουμε ότι η B είναι εντός της A.

$$a \cap b = b$$

Επίσης θα μπορούσαμε να πούμε ότι η A περιέχει την B χρησιμοποιώντας την αντίστροφη σχέση contains.



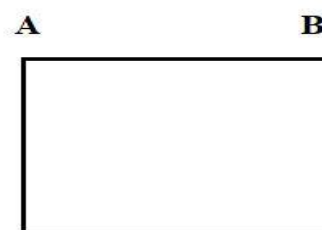
Εικόνα 11- Within-Contains

Equals

Οι γεωμετρίες είναι τοπολογικά ίσες, δηλαδή, τα εσωτερικά τους έχουν τουλάχιστον ένα κοινό σημείο και κανένα σημείο του εσωτερικού ή του ορίου της μίας γεωμετρίας δεν έχει κοινό σημείο με το εξωτερικό της άλλης.

$$(a \cap b = a) \wedge (a \cap b = b)$$

Μια εναλλακτική περιγραφή δίνεται από την παραπάνω σχέση όπου η τομή των συνόλων a,b είναι ταυτόχρονα και το σύνολο a και το σύνολο b.



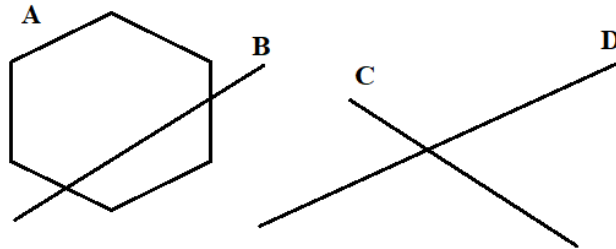
Εικόνα 12 – Equals

Crosses

Οι γεωμετρίες έχουν κάποια, αλλά όχι όλα, εσωτερικά σημεία κοινά και η διάσταση της γεωμετρίας που σχηματίζουν τα κοινά τους σημεία είναι μικρότερη από τη διάσταση τουλάχιστον μίας από τις δύο αρχικές γεωμετρίες (A,B). Εναλλακτικά πρέπει η διάσταση οποιασδήποτε από τις δύο γεωμετρίες (ή και των δύο) να είναι 1 (C,D).

$$\dim(a) \neq \dim(b) \text{ or } \dim(\text{any}) = 1$$

Στην παρακάτω εικόνα βλέπουμε τις απεικονίσεις των προαναφερθεισών περιπτώσεων.



Εικόνα 13 – Crosses

Overlaps

Οι γεωμετρίες έχουν την ίδια διάσταση και κάποια κοινά σημεία, αλλά όχι όλα. Επίσης, η γεωμετρία που δημιουργούν τα κοινά σημεία των δύο γεωμετριών πρέπει να έχει ίδια διάσταση με τις δύο αρχικές γεωμετρίες.

$$\dim(a) = \dim(b)$$

Άρα όταν αγγίζονται μόνο οι ακμές δύο γεωμετριών και όταν μια γεωμετρία είναι ολόκληρη εντός μιας άλλης δεν πληρούνται οι προϋποθέσεις της overlaps.

2.3.2 Σχέσεις απόστασης

Εκτός από τις σχέσεις θέσης που αναφέρθηκαν παραπάνω, δύο γεωμετρίες μπορούν να συσχετιστούν και με βάση την απόσταση. Συγκρίνοντας την απόσταση μίας γεωμετρίας από πολλές άλλες, μπορεί να βρεθεί η πιο κοντινή (nearest) και η πιο μακρινή (furthest), βρίσκοντας την ελάχιστη και τη μέγιστη απόσταση αντίστοιχα.

Επίσης, μπορεί να ακολουθηθεί κάποια πιο γενική προσέγγιση, ορίζοντας διαστήματα αποστάσεων και τις αντίστοιχες ερμηνείες τους.

Υπάρχουν διάφορες προσεγγίσεις του θέματος της απόστασης, αλλά στα πλαίσια της εργασίας αυτής θα χρησιμοποιηθεί η Ευκλείδεια απόσταση, ο τύπος υπολογισμού της οποίας έχει αναφερθεί παραπάνω.

2.3.3 Σχέσεις κατεύθυνσης

Οι σχέσεις κατεύθυνσης δείχνουν τον τρόπο με τον οποίο είναι τοποθετημένο ένα αντικείμενο σε σχέση με ένα άλλο αντικείμενο αναφοράς. Χωρίζονται σε εσωτερικές, για αντικείμενα που βρίσκονται εντός του αντικειμένου αναφοράς και εξωτερικές, για αντικείμενα που βρίσκονται εκτός του αντικειμένου αναφοράς. Κάποια παραδείγματα εσωτερικών σχέσεων κατεύθυνσης είναι: αριστερά, στο κέντρο, δεξιά και κάτω κλπ. Αντίστοιχα για τις εξωτερικές σχέσεις κατεύθυνσης: στα δεξιά του, μπροστά, πίσω, απο πάνω κλπ.

2.3.4 Σχέσεις με βάση την κατηγορία

Όπως έχει αναφερθεί, κάθε χωρικό αντικείμενο στην δισδιάστατη μοντελοποίηση μπορεί να κατηγοριοποιηθεί ως σημείο, ευθύγραμμο τμήμα-διάνυσμα ή πολύγωνο. Σύμφωνα με αυτά, μπορούν να οριστούν σχέσεις με βάση τις κατηγορίες των συμμετεχόντων αντικειμένων. Αυτές είναι σχέσεις:

- σημείο-σημείο
- σημείο-διάνυσμα
- σημείο-πολύγωνο
- διάνυσμα-διάνυσμα
- διάνυσμα-πολύγωνο
- πολύγωνο-πολύγωνο

2.3.5 Χώρο-χρονικές σχέσεις

Οι χώρο-χρονικές (spatio-temporal) σχέσεις δείχνουν πως μεταβάλλονται όλες οι προηγούμενες σχέσεις όταν προστεθεί στην εξίσωση ο παράγοντας του χρόνου. Αυτό μπορεί να είναι βραχυπρόθεσμα (κίνηση οχημάτων, πλοίων, αεροσκαφών) ή

μακροπρόθεσμα (δημιουργία/καταστροφή κτιρίων, μετακίνηση μνημείων, αλλαγές συνόρων κλπ).[8]

2.3.6 WKT και γεωμετρικά αντικείμενα

Η WKT (Well Known Text) είναι μια γλώσσα σήμανσης που αναπαριστά διανυσματικά γεωμετρικά αντικείμενα πάνω σε χάρτη, καθώς επίσης και συστήματα συντεταγμένων και μετασχηματισμούς μεταξύ τους. Χρησιμοποιείται ευρέως σε εφαρμογές που διαχειρίζονται χωρικά δεδομένα αφού αναπαριστά όλες τις γεωμετρίες με τρόπο που είναι εύκολα αναγνώσιμος.

Η βασική της σύνταξη έχει ως εξής:

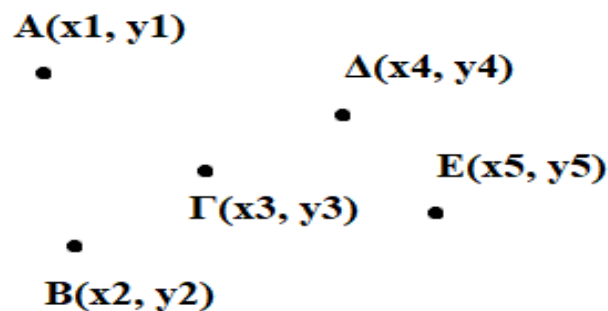
Για σημεία: POINT (x y)

Για διανύσματα: LINESTRING (x1 y1, x2 y2, ..., xn yn)

Για πολύγωνα: POLYGON (x1 y1, x2 y2, x3 y3 ..., x1 y1)

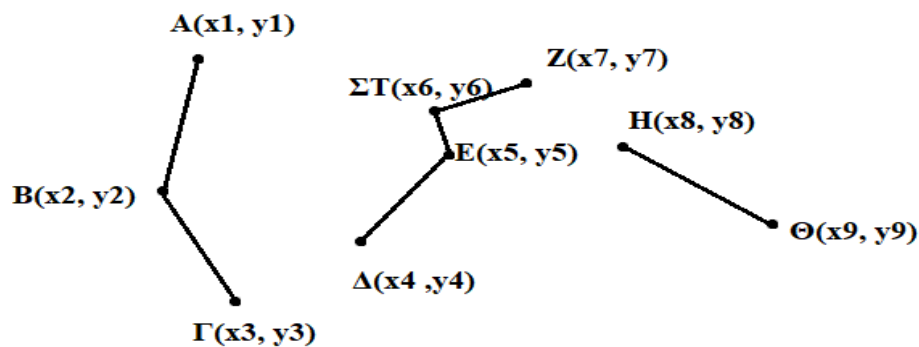
Φυσικά, μία γεωμετρία ενδέχεται να περιέχει παραπάνω από ένα γεωμετρικό αντικείμενο ίδιου τύπου ή αντικείμενα διαφορετικών τύπων.

Για παραπάνω από ένα σημεία έχουμε: MULTIPOINT((x1 y1), (x2 y2), ... (xn yn))



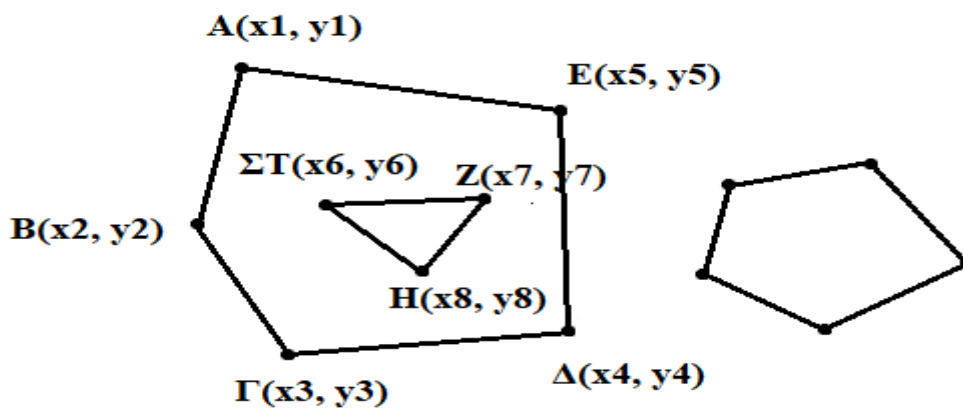
Εικόνα 14 – MultiPoint

Για παραπάνω από ένα ευθύγραμμα τμήματα έχουμε: MULTILINESTRING ((x1 y1, x2 y2, ..., xn yn), (x3 y3, x4 y4, ..., xm ym), (xk yk, ..., xl yl))



Εικόνα 15 – MultiLineString

Για παραπάνω απο ένα πολύγωνα ισχύει: MULTIPOLYGON((x1 y1, x2 y2, x3 y3 ..., x1 y1), (x4 y4, x5 y5, ..., x4 y4),, (xn yn, ..., xn yn))\



Εικόνα 16 – MultiPolygon

Τέλος, για αντικείμενα διαφορετικών τύπων έχουμε: GEOMETRYCOLLECTION(POINT (x y), LINESTRING (x1 y1, x2 y2, ..., xn yn), POLYGON (x1 y1, x2 y2, x3 y3 ..., x1 y1)) με όποια σειρά και επαναλήψεις είναι αναγκαίες για να αναπαρασταθεί αυτό που πρέπει.



Εικόνα 17 – GeometryCollection

Πληθώρα γλωσσών προγραμματισμού περιέχουν μεθόδους για την ανάγνωση της γλώσσας WKT και για την δημιουργία γεωμετρικών αντικειμένων κατόπιν. Ο απλός και σαφής τρόπος αναπαράστασης που προσφέρει διευκολύνει σε μεγάλο βαθμό την διαχείριση των χωρικών δεδομένων και την χρήση τους για οποιονδήποτε σκοπό κριθεί απαραίτητο.

2.4 OpenStreetMap και πηγές χωρικών δεδομένων

Μια έρευνα στο διαδίκτυο θα επιστρέψει πολλές πηγές ανοιχτών ή δωρεάν δεδομένων που παρέχονται για κάποιο σκοπό και μετά από μια συμφωνία με τον πάροχό τους, στην Ελλάδα και διεθνώς.

Το OpenStreetMap (OSM) είναι ένα συνεργατικό εγχείρημα με στόχο την δημιουργία ενός δωρεάν και τροποποιήσιμου χάρτη του κόσμου. Τελικά, εκτός του χάρτη αυτού καθ'αυτού, το σημαντικότερο προϊόν της πρωτοβουλίας αυτής, είναι τα δεδομένα που παράγει. Έμπνευση για την πραγμάτωση της στάθηκε η έλλειψη οικονομικών συσκευών δορυφορικής πλοήγησης και οι περιορισμοί διαθεσιμότητας γεωγραφικών πληροφοριών σε μεγάλο μέρος του κόσμου.

Δημιουργήθηκε από τον Steve Coast το 2004 και από τότε έχει αναπτυχθεί σε σημείο να έχει πάνω από δύο εκατομμύρια εγγεγραμμένους χρήστες/εθελοντές. Οι χρήστες αυτοί μπορούν να συλλέγουν δεδομένα χρησιμοποιώντας χειρονακτική τοπογράφιση, συσκευές GPS, αεροφωτογραφίες, ψηφιακές κάμερες και άλλες δωρεάν πηγές. Τα δεδομένα που συλλέγονται, εισάγονται στην OSM βάση δεδομένων και είναι διαθέσιμα σε όλους κάτω από την Open Database License (ODbL).

Για τις ανάγκες της εφαρμογής που αναπτύξαμε, χρησιμοποιήσαμε πηγές χωρικών/γεωγραφικών δεδομένων από το OpenStreetMap. Η πρώτη πηγή περιέχει δεδομένα που αναπαριστούν το οδικό δίκτυο (road network) όλης της Ελλάδας χρησιμοποιώντας ευθύγραμμα τμήματα/διανύσματα και η δεύτερη περιέχει δεδομένα που αντιπροσωπεύουν σημεία ενδιαφέροντος (points of interest) με τη μορφή πολυγώνων.

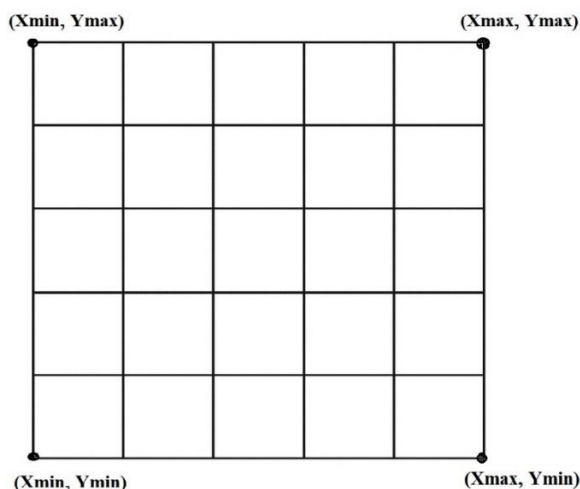
3

Προσέγγιση προβλήματος και περιγραφή αλγορίθμων

Στο παρών κεφάλαιο θα γίνει επεξήγηση του τρόπου που προσεγγίστηκε το πρόβλημα, αναφορά στις δομές δεδομένων που χρησιμοποιήθηκαν και ανάλυση των αλγορίθμων ολοκλήρωσης χωρικών δεδομένων που αναπτύχθηκαν.

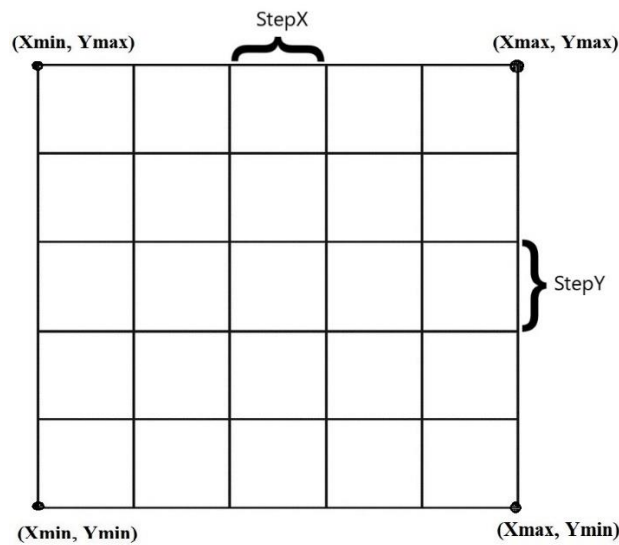
3.1 Πλέγμα

Ο τρόπος που χρησιμοποιήθηκε για να αντιμετωπιστεί το πρόβλημα του υψηλού αλγοριθμικού κόστους και χρόνου εκτέλεσης που προκύπτει από την σύγκριση κάθε γεωμετρικού αντικειμένου με όλα τα υπόλοιπα, είναι η δημιουργία ενός πλέγματος (grid). Ουσιαστικά, όλη η χωρική περιοχή που έχει επιλεγεί, διαιρείται σε έναν πεπερασμένο αριθμό ορθογωνίων κελιών, που ορίζεται από εμάς. Προφανώς, όσο μικρότερο είναι το πλήθος των κελιών, τόσο μεγαλύτερο θα είναι το μέγεθος τους, και αντίστροφα.



Εικόνα 18 - Πλέγμα

Αρχικά, πρέπει να ελεγχθούν όλα τα σημεία των χωρικών δεδομένων, μία φορά, ώστε να προσδιοριστούν οι ελάχιστες και οι μέγιστες τετμημένες και τεταγμένες. Όπως φαίνεται στην παραπάνω εικόνα, τα τέσσερα σημεία που προκύπτουν από τις προαναφερθείσες συντεταγμένες, αποτελούν τις γωνίες του πλέγματος. Στη συνέχεια, αφού έχει επιλεγθεί ένας αριθμός n που αντιπροσωπεύει τον αριθμό των κελιών σε κάθε πλευρά του πλέγματος, και άρα υπάρχουν $n \times n$ κελιά συνολικά, πρέπει να προσδιοριστεί το βήμα (step) των x και y , δηλαδή ποια είναι η απόσταση μεταξύ των 2 γωνιών κάθε κελιού στον άξονα των τετμημένων και στον άξονα των τεταγμένων αντίστοιχα.



Εικόνα 19 – Καθορισμός μεγέθους κελιών με βάση τα βήματα x και y

Το $StepX$, δηλαδή το βήμα στον άξονα των τετμημένων, προκύπτει αν διαιρεθεί το μήκος της πλευράς του grid στον άξονα των x με το πλήθος των κελιών στην πλευρά αυτή. Άρα,

$$StepX = \frac{(Xmax - Xmin)}{n}$$

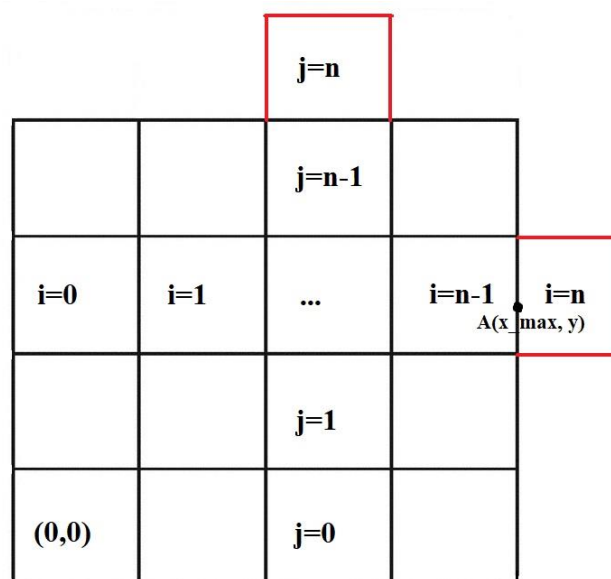
Και αντίστοιχα το βήμα στον άξονα των τεταγμένων, $StepY$,

$$StepY = \frac{(Ymax - Ymin)}{n}$$

Έχοντας υπολογίσει τις παραπάνω πληροφορίες, είναι πλέον δυνατό να βρεθεί σε ποιο κελί του πλέγματος βρίσκεται οποιοδήποτε σημείο. Εστω σημείο $A(x,y)$, τότε προσδιορίζεται το κελί (i,j) που ανήκει ως εξής:

$$j = \frac{(y - Y_{min})}{StepY}$$

Έστω ένα σημείο που βρίσκεται στο πρώτο κελί του άξονα των τετμημένων. Κάνοντας χρήση του παραπάνω τύπου για να βρεθεί το i , βλέπουμε ότι ο αριθμητής είναι μικρότερος από το StepX και άρα μένει 0 στο ακέραιο μέρος. Αφού η αρίθμηση των κελιών ξεκινάει από το μηδέν, πρέπει κανονικά να πηγαίνει έως το $n-1$. Αν, πχ έχει επιλεγεί ως n το 50, το πρώτο κελί σε οποιονδήποτε από τους δύο άξονες θα έχει τον αριθμό 0 ως i ή j και το τελευταίο τον αριθμό 49 αντίστοιχα. Υπάρχει όμως μια ειδική περίπτωση η οποία πρέπει να εξεταστεί. Εάν στους παραπάνω τύπους για τα i και j , αντικατασταθεί $x=X_{\max}$ και $y=Y_{\max}$ αντίστοιχα, θα προκύψει $i=n$ και $j=n$ αντί για $n-1$. Όπως φαίνεται στην παρακάτω εικόνα, τα κελιά αυτά είναι εκτός του πλέγματος και άρα δεν περιέχουν γεωμετρίες.



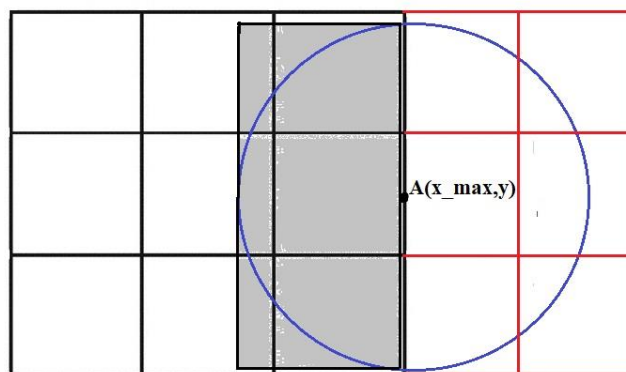
32

Για να αποφευχθεί λοιπόν η αναζήτηση γεωμετρικών αντικειμένων σε άδεια, εκτός πλέγματος κελιά στην ειδική περίπτωση που αναφέρθηκε, προστέθηκαν οι εξής έλεγχοι:

if ($i \geq n$) *then* $i = n - 1$;

if ($j \geq n$) *then* $j = n - 1$;

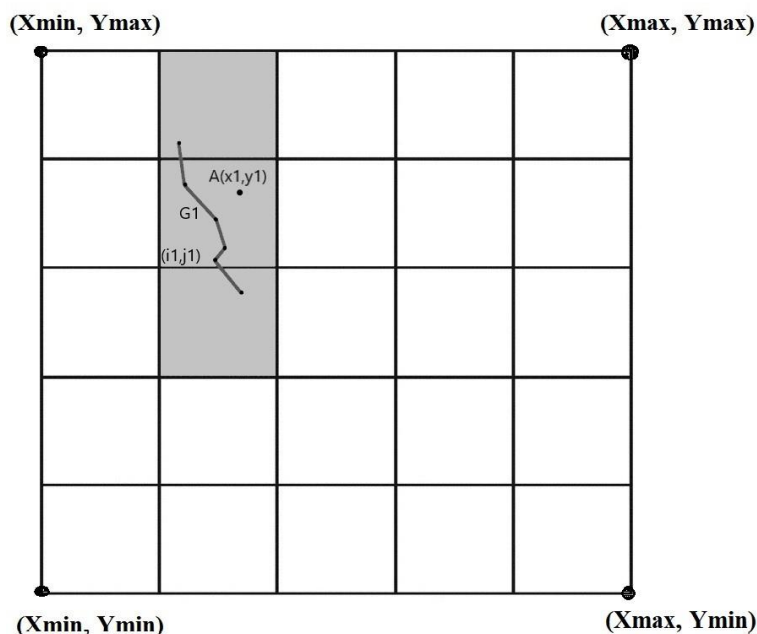
Για την αναζήτηση κοντινότερου δρόμου, θα επιλεγθεί το αμέσως προηγούμενο κελί, με αρίθμηση $n-1$, δηλαδή το τελευταίο του πλέγματος στον άξονα των τετμημένων αν το i ήταν ίσο ή μεγαλύτερο του n , και το τελευταίο στον άξονα των τεταγμένων αν το j ήταν μεγαλύτερο ή ίσο του n . Ο λόγος που στους παραπάνω ελέγχους εξετάζονται και τιμές των i, j μεγαλύτερες του n , και όχι μόνο για ισότητα είναι η δεύτερη λειτουργία του αλγορίθμου που αναπτύχθηκε. Όταν ο αλγόριθμος αναζητά τα σημεία ενδιαφέροντος εντός ενός κύκλου ενδέχεται ένα μέρος του κύκλου αυτού να βρίσκεται εκτός του πλέγματος. Το αν θα συμβεί αυτό και σε τι βαθμό, εξαρτάται από το κέντρο και την ακτίνα του κύκλου, υπάρχει όμως η περίπτωση το μέρος αυτό να τέμνει ορισμένα υποθετικά, άδεια κελιά (απεικονίζονται με κόκκινο) τα οποία έχουν i ή j μεγαλύτερο του n , εκτός από αυτά που έχουν i ή j ίσο με n . Σε οποιαδήποτε περίπτωση, οι έλεγχοι διασφαλίζουν ότι ο αλγόριθμος θα ελέγξει το κομμάτι του κύκλου που βρίσκεται εντός του grid. Ο τρόπος με τον οποίο διαπιστώνεται ποια κελιά τέμνει ο κύκλος αναλύεται παρακάτω.



Εικόνα 21– Κέντρο κύκλου στο όριο του πλέγματος

Παραπάνω απεικονίζεται η ακραία περίπτωση που το κέντρο του κύκλου βρίσκεται στο όριο του πλέγματος, το κομμάτι του κύκλου και τα κελιά που τέμνει εντός του πλέγματος (γκρι) και το κομμάτι εκτός του.

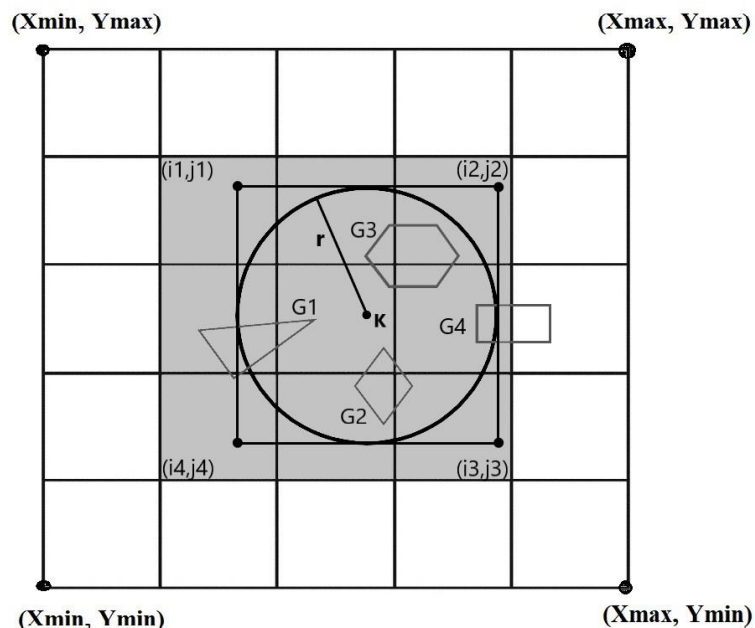
Αφού είναι γνωστό σε ποιο κελί του πλέγματος ανήκει ένα σημείο, πρέπει να γίνουν οι απαραίτητες συγκρίσεις μόνο με τις γεωμετρίες που ανήκουν σε αυτό, ολικά ή μερικά.



Εικόνα 22 – Αναπαράσταση MultiLineString σε πλέγμα

Όσον αφορά τη σχέση nearest για σημεία και δρόμους, στην παραπάνω εικόνα φαίνεται ότι το σημείο A ανήκει στο κελί $(i1,j1)$ και άρα πρέπει να προσδιοριστεί ποιες γεωμετρίες ανήκουν σε αυτό, όπως η G1. Αυτό θα γίνει, εξετάζοντας τα σημεία που τις απαρτίζουν, και αν έστω ένα ανήκει στο εν λόγω κελί, η γεωμετρία συμπεριλαμβάνεται στη λίστα αυτών που ανήκουν στο κελί και συνεπώς θα εξεταστεί η απόσταση τους από το σημείο. Η γεωμετρία που αντιστοιχεί στην μικρότερη απόσταση που θα βρεθεί, είναι αυτή που ψάχνουμε.

Για την σχέση intersects μεταξύ πολυγώνων και ενός κύκλου με κέντρο και ακτίνα που έχει οριστεί από εμάς, και άρα για να προσδιοριστεί ποια σημεία ενδιαφέροντος βρίσκονται εντός μιας συγκεκριμένης εμβέλειας από κάποιο σημείο, αρχικά χρησιμοποιείται το τετράγωνο που εγγράφει τον κύκλο.



Εικόνα 23 – Αναπαράσταση εγγεγραμμένου κύκλου και πολυγώνων σε πλέγμα

Όπως φαίνεται στην παραπάνω εικόνα, έχοντας τον κύκλο με κέντρο K και ακτίνα r , ανακαλύπτεται το τετράγωνο που τον εγγράφει και άρα τα κελιά στα οποία βρίσκονται τα τέσσερα σημεία που είναι οι γωνίες του τετραγώνου. Παίρνοντας όλους τους δυνατούς συνδυασμούς από τις ελάχιστες και μέγιστες τετμημένες και τεταγμένες, βρίσκονται όλα τα κελιά που περιέχει το τετράγωνο και άρα ο κύκλος. Τέλος, ο κύκλος συγκρίνεται με όλες τις γεωμετρίες που περιέχουν τα εν λόγω κελιά και επιστρέφονται όσες πλοιρούν την σχέση intersects, δηλαδή έχουν τουλάχιστον ένα κοινό σημείο με τον κύκλο.

Για να είναι δυνατά όμως όλα τα παραπάνω, πρέπει να είναι γνωστό ποιές γεωμετρίες ανήκουν σε κάθε κελί του grid και η πληροφορία αυτή να είναι διαθέσιμη στον αλγόριθμο ανα πάσα στιγμή. Για το λόγο αυτό, είναι απαραίτητες κάποιες κατάλληλες δομές δεδομένων, οι οποίες αναλύονται παρακάτω.

3.2 Δομές δεδομένων

Στην πληροφορική, η έννοια της δομής δεδομένων αναφέρεται στους ποικίλους δυνατούς τρόπους οργάνωσης και αποθήκευσης δεδομένων μέσα σε έναν υπολογιστή, ώστε να μπορεί να επιτευχθεί αποδοτική προσπέλαση και τροποποίηση τους. Στον ορισμό της δομής δεδομένων εμπεριέχεται η εισαγωγή και απομάκρυνση στοιχείων από αυτή με τρόπο που δεν αλλοιώνεται η κατασκευή της. Οι δομές

δεδομένων χρησιμοποιούνται ευρέως στην δημιουργία αλγορίθμων λόγω της χρησιμότητάς τους στην διαχείριση μεγάλου όγκου δεδομένων. Διαφορετικές δομές ενδείκνυνται για διαφορετικές εφαρμογές και κάποιες είναι εξειδικευμένες σε μεγάλο βαθμό, ώστε να έχουν την αποδοτικότερη δυνατή χρήση σε συγκεκριμένες υλοποιήσεις.

3.2.1 Πίνακες

Πίνακας (Array) ονομάζεται η δομή δεδομένων η οποία χρησιμοποιεί έναν αριθμητικό δείκτη για να προσδιορίσει τη θέση των στοιχείων που περιέχει εντός της. Οι πίνακες είναι από τις παλαιότερες και σημαντικότερες δομές δεδομένων και χρησιμοποιούνται σχεδόν σε κάθε πρόγραμμα με κάποια μορφή. Επίσης, χρησιμοποιούνται για να υλοποιηθούν άλλες δομές όπως πχ λίστες.

2	14	62	35	7
0	1	2	3	4

Εικόνα 24 – Πίνακας

Στην παραπάνω οπτική αναπαράσταση, ο πίνακας έχει πέντε θέσεις. Στην πρώτη θέση, που έχει τον δείκτη 0, βρίσκεται ο αριθμός 2, στην δεύτερη που έχει δείκτη 1 βρίσκεται ο αριθμός 14 κ.ο.κ. Ο πίνακας αριθμείται ξεκινώντας από το 0 και όχι από το 1. Αυτό αποτελεί μια κοινή τακτική αρίθμησης για την πλειοψηφία των σύγχρονων γλωσσών προγραμματισμού, όπως είναι η Java, C++ κλπ. Έτσι, δοθέντος ενός πίνακα N θέσεων, μπορούν να προσπελαστούν τα στοιχεία 0 μέχρι N-1. Αναζήτηση του στοιχείου στη θέση N θα οδηγούσε σε προγραμματιστικό λάθος (κατά κανόνα) κατά την εκτέλεση.

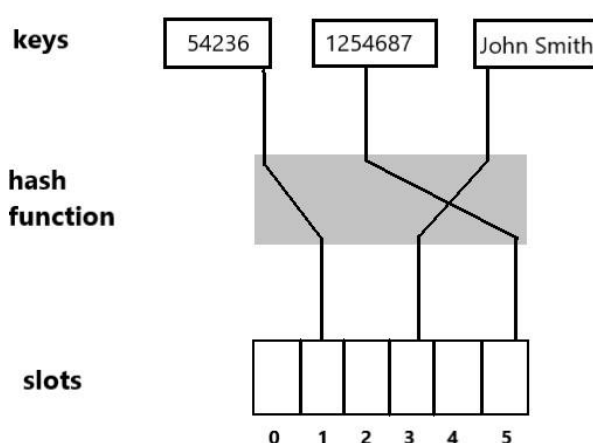
Άλλη μία σημαντική ιδιότητα των πινάκων είναι ότι μπορεί να χρειάζεται πάνω από έναν δείκτη για να προσδιοριστεί και από τον αριθμό των δεικτών προκύπτει και η διάσταση του πίνακα. Αν έχει έναν είναι μονοδιάστατος και αναπαρίσταται οπτικά από μία μόνο γραμμή ή στήλη, αν έχει δύο είναι δυοδιάστατος και αναπαρίσταται ως τετράγωνο, αν έχει τρεις ως κύβος κλπ.

Επίσης, υπάρχουν στατικοί πίνακες, των οποίων το μέγεθος ορίζεται κατά την αρχικοποίηση και παραμένει σταθερό, και δυναμικοί (dynamic arrays ή array lists), οι

οποίοι δεν έχουν σταθερό μέγεθος και αυτό μεταβάλλεται με την πρόσθεση ή αφαίρεση στοιχείων στον και από τον πίνακα.

3.2.2 Πίνακες κατακερματισμού

Πίνακας κατακερματισμού (Hash Table) ονομάζεται μια δομή δεδομένων η οποία αντιστοιχίζει κλειδιά σε στοιχεία. Η κατασκευή της βασίζεται στη δομή του πίνακα που αναφέραμε παραπάνω και σε μια συνάρτηση κατακερματισμού (hash function). Στην περίπτωση του πίνακα, ένα στοιχείο με κλειδί **k** καταχωρείται στην θέση **k**, αν το κλειδί είναι αριθμητικού τύπου και κάνουμε την παραδοχή ότι είναι όμοιο με τον δείκτη της θέσης που κατέχει το αντίστοιχο στοιχείο του στον πίνακα. Σε διαφορετική περίπτωση, για να βρεθεί ποιο στοιχείο αντιστοιχεί σε ένα κλειδί, θα έπρεπε να χρησιμοποιηθούν δύο πίνακες. Ο ένας θα περιείχε τα κλειδιά και ο άλλος τα στοιχεία σε θέσεις με τον ίδιο δείκτη, και θα έπρεπε να γίνει αναζήτηση στον πρώτο χρησιμοποιώντας τόσους βρόχους όσες οι διαστάσεις του, μέχρι να βρεθεί σε ποια θέση είναι το επιθυμητό κλειδί. Όσο το μέγεθος των πινάκων όμως μεγαλώνει το κόστος μιας τέτοιας προσέγγισης γίνεται ασύμφορο. Για τον λόγο αυτό χρησιμοποιείται η συνάρτηση κατακερματισμού. Το κλειδί δίνεται στην συνάρτηση η οποία με βάση αυτό, υπολογίζει έναν δείκτη **i** που αντιπροσωπεύει την θέση αποθήκευσης του στοιχείου στον πίνακα. Άρα ισχύει $h(k)=i$. Η συνάρτηση κατακερματισμού είναι ντετερμινιστική, δηλαδή για το ίδιο κλειδί θα βγάλει πάντα τον ίδιο δείκτη.



Εικόνα 25- Πίνακας κατακερματισμού

Υπάρχει όμως περίπτωση, η συνάρτηση να καταλήξει στο ίδιο αποτέλεσμα με παραπάνω από ένα κλειδιά και τότε λέμε ότι έχουμε σύγκρουση. Φυσικά, το ιδανικό

θα ήταν οι συγκρούσεις να μπορούσαν να αποφευχθούν εξ ολοκλήρου, αλλά στην πράξη κάτι τέτοιο δεν είναι εφικτό. Με την επιλογή μιας καλά σχεδιασμένης συνάρτησης, οι συγκρούσεις ελαχιστοποιούνται, όμως είναι αναγκαίο όταν συμβαίνουν, να μπορούν να με κάποιο τρόπο να αντιμετωπίζονται. Έχουν αναπτυχθεί διάφορες μέθοδοι αντιμετώπισης των συγκρούσεων με πιο συνήθη την chaining, η οποία χρησιμοποιεί συνδεδεμένες λίστες για να αποθηκεύει τα στοιχεία στην περίπτωση σύγκρουσης, όπου η θέση του πίνακα χρησιμοποιείται ήδη από το στοιχείο του πρώτου κλειδιού.[4][5][6]

3.3 Αλγόριθμοι ολοκλήρωσης χωρικών δεδομένων

Παρακάτω θα αναλυθεί η λειτουργία δύο αλγορίθμων ολοκλήρωσης χωρικών δεδομένων, τόσο ενός απλού brute force αλγορίθμου όσο και του αλγορίθμου βασισμένο σε πλέγμα και πίνακες κατακερματισμού, που αναπτύχθηκαν στα πλαίσια της εργασίας αυτής.

3.3.1 Brute force αλγόριθμος

Για την σχέση nearest, ο αλγόριθμος αυτός βρίσκει την Ευκλείδεια απόσταση του σημείου από κάθε δρόμο και κρατάει την μικρότερη. Για το επόμενο σημείο πρέπει να κάνει ακριβώς το ίδιο και αυτό θα επαναληφθεί για όσα σημεία έχουμε. Όσον αφορά την σχέση intersects, πρέπει για κάθε κύκλο να ελέγξει όλα τα υπάρχοντα πολύγωνα και να επιστρέψει όσα πλοιοούν την σχέση και αυτό να γίνει για όλους τους κύκλους. Είναι φανερό ότι υπάρχει δραματική αύξηση του αλγοριθμικού κόστους όσο αυξάνεται ο αριθμός σημείων και κύκλων για έλεγχο.

3.3.2 Grid-based αλγόριθμος

Αφού περιγράφηκε η λειτουργία του πλέγματος και παρατέθηκαν οι απαραίτητες πληροφορίες για την κατανόηση των δομών δεδομένων που χρησιμοποιήθηκαν, μπορεί πλέον να αναλυθεί λεπτομερώς ο αλγόριθμος ολοκλήρωσης χωρικών δεδομένων που αναπτύχθηκε. Ο αλγόριθμος αυτός, σε αντίθεση με τον brute force, χρειάζεται να προσπελάσει τα χωρικά δεδομένα μία φορά ώστε να δημιουργηθεί το πλέγμα και άλλη μία για να γεμίσουν οι πίνακες κατακερματισμού με βάση αυτό. Από εκεί και μετά, για όσα σημεία και κύκλους υπάρχουν στα δεδομένα μας, το κόστος για να ανακαλυφθούν οι σχέσεις nearest και intersects είναι πολύ μικρό, αφού

χρειάζεται να γίνουν συγκρίσεις μόνο με τις γεωμετρίες που υπάρχουν στο κελί ή κελιά που βρίσκεται το κάθε σημείο και κύκλος αντίστοιχα.

```
while( roads source has more lines) {  
    extract coordinates from geometry String;  
    find minX, maxX, minY, maxY;  
}
```

Όπως έχει αναφερθεί, ο κάθε δρόμος αναπαρίσταται από σημεία ενωμένα με ευθύγραμμα τμήματα. Ο αλγόριθμος αρχικά εξετάζει τα σημεία του κάθε δρόμου ώστε να ανακαλύψει τις συντεταγμένες που χρειάζονται για την δημιουργία του grid, όπως έχει αναλυθεί στην αντίστοιχη ενότητα.

```
CreateGrid(minX, maxX, minY, maxY, n);  
CreateRoadHashTable(key, ArrayList(roads));  
CreatePOIHashTable(key, ArrayList(pois));
```

Το πλέγμα δημιουργείται με συνολικό αριθμό κελιών $n \times n$ που ορίζεται από εμάς. Στη συνέχεια δημιουργούνται δύο πίνακες κατακερματισμού. Και οι δύο περιέχουν δυναμικούς πίνακες (στο εξής θα αναφερόμαστε σε αυτούς ως ArrayList), αφού δεν είναι γνωστό εξ αρχής πόσα στοιχεία θα περιέχει ο καθένας, και ο καθένας αντιστοιχίζεται με ένα κλειδί. Αρχικά οι πίνακες κατακερματισμού είναι άδειοι και το πόσα ArrayList θα περιέχει τελικά ο καθένας και πόσα αντικείμενα το κάθε ArrayList θα εξαρτηθεί από τις πηγές δεδομένων που θα χρησιμοποιηθούν.

```
/*Fill Hash Tables*/  
while(source has more lines) {  
    extract coordinates from geometry String;  
    table[] ← coordinates ;  
    for(i=0; i < table.length; i = i + 2){  
        (i,j) ← findGridCell(table[i], table[i+1]);  
        key ← 'ij';  
        if(previous point's cell is different from current){  
            if(hashtable contains key) {  
                hashtable.get(key).add(geolD object);  
            } else {  
                create ArrayList;  
                arrayList.add(geolD object);  
                hashtable.put(key, ArrayList);  
            } else {
```

```

        continue;
    }
}

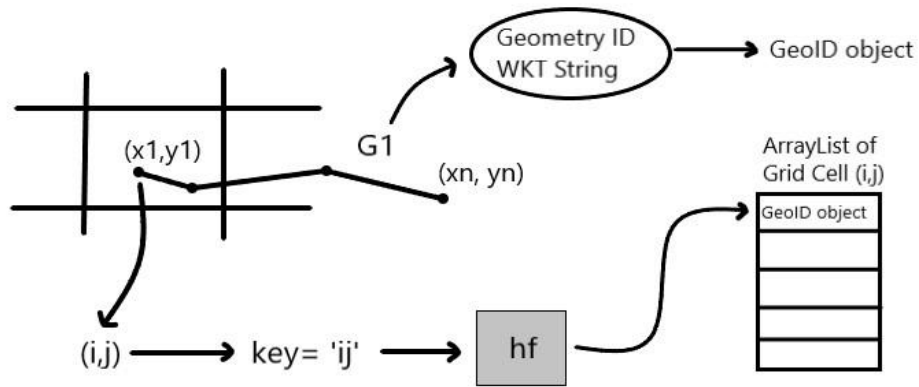
```

Με τον παραπάνω τρόπο γεμίζει ο κάθε πίνακας κατακερματισμού. Ο κώδικας αυτός θα τρέξει μία φορά για την πηγή που περιέχει τους δρόμους και μία φορά για την πηγή των σημείων ενδιαφέροντος. Απο κάθε γραμμή της κάθε πηγής χωρικών δεδομένων δημιουργείται αρχικά ένας πίνακας με την εξής δομή:

x1	y1	x2	y2	x _{v-1}	y _{v-1}	x _v	y _v
----	----	----	----	-----	-----	------------------	------------------	----------------	----------------

Οι άρτιες θέσεις του πίνακα περιέχουν τις τετμημένες των σημείων που απαρτίζουν την γεωμετρία και οι περιττές, τις τεταγμένες. Τα σημεία ελέγχονται σειριακά από τον αλγόριθμο. Για κάθε σημείο προσδιορίζεται σε ποιο κελί του πλέγματος ανήκει και στην συνέχεια απο τα i, j που αντιπροσωπεύουν το κελί αυτό, δημιουργείται ένα κλειδί.

Εαν το κελί που βρέθηκε είναι το ίδιο με του προηγούμενου σημείου, ο αλγόριθμος, προχωράει στο επόμενο σημείο χωρίς να κάνει κάτι. Αυτό γίνεται διότι αν το ίδιο κελί έχει παρουσιαστεί ξανά για την ίδια γεωμετρία, έχουμε ήδη προσθέσει ήδη την γεωμετρία αυτή στο ArrayList που αντιστοιχεί στο κελί που βρέθηκε οτι ανήκει το σημείο. Αξίζει να σημειωθεί οτι τα ArrayList περιέχουν αντικείμενα τα οποία απαρτίζονται από το id και το WKT String που περιγράφει την κάθε γεωμετρία και τα WKT String μετατρέπονται σε γεωμετρικά αντικείμενα μόνο στην περίπτωση που χρειάζεται να γίνουν συγκρίσεις. Εάν είναι διαφορετικό, ελέγχεται αν ο πίνακας κατακερματισμού περιέχει το κλειδί και άρα υπάρχει ήδη ArrayList απο σημεία άλλων γεωμετριών που ανήκουν στο ίδιο κελί και έχουν ελεγχθεί νωρίτερα. Αν το περιέχει, προστίθεται η γεωμετρία σε αυτό. Αν όχι, δημιουργείται ένα νέο ArrayList, προστίθεται η γεωμετρία σε αυτό και στη συνέχεια προστίθεται το ίδιο στον πίνακα κατακερματισμού αντιστοιχισμένο με το κλειδί.



Εικόνα 26 – Γέμισμα πίνακα κατακερματισμού

Αφού οι πίνακες κατακερματισμού έχουν γεμίσει και η πληροφορία για το ποιές γεωμετρίες ανήκουν σε κάθε κελί, είναι πλέον άμεσα διαθέσιμη, μπορούν να εντοπιστούν οι επιθυμητές σχέσεις.

```

/*Find nearest road for each point*/
while(points source has more lines){
    (i,j) ← point.findGridCell();
    key ← 'ij';
    RoadList ← RoadHashTable.get(key);
    minDistance ← very big number;
    if(RoadList == null){
        for(k=i-C; k <= i+C; k++){
            for(l=j-C; l <= j+C; l++){
                key2 ← 'kl'
                RoadList2 ← RoadHashTable.get(key2);
                if(RoadList2 != null){
                    for(m=0; m < RoadList2.length; m++) {
                        distance ← point.distanceFrom(RoadList2[m]);
                        if(distance < minDistance) {
                            minDistance ← distance;
                            road_id ← RoadList[m].getID;
                        }
                    }
                }
            }
        }
    }
} else {
    for(i=0; i < RoadList.length; i++) {

```

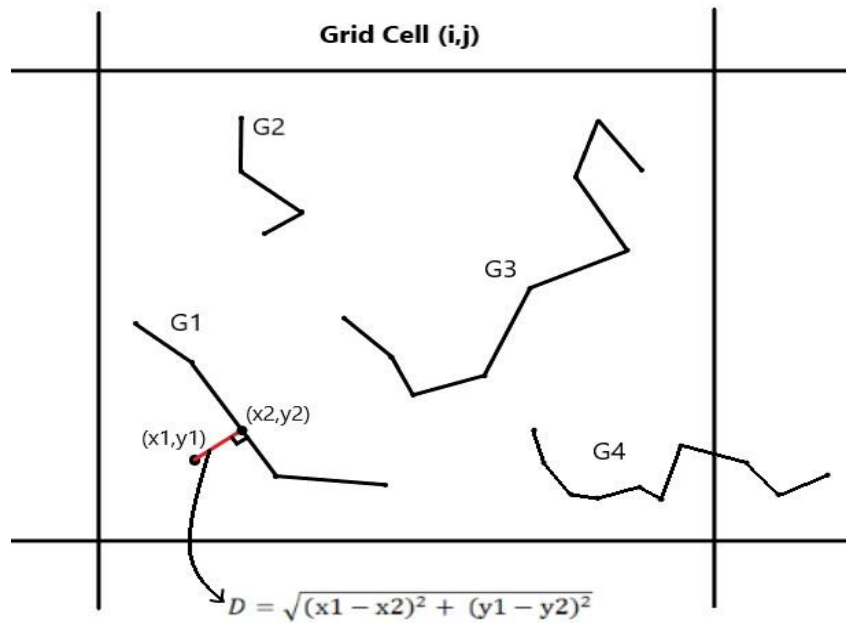
```

        distance ← point.distanceFrom(RoadList[i]);
        if(distance < minDistance) {
            minDistance ← distance;
            road_id ← RoadList[i].getID;
        }
    }
}
return road_id, minDistance;
}

```

Για την σχέση nearest, για κάθε σημείο που έχει οριστεί αρχικά προσδιορίζεται σε ποιο κελί του grid ανήκει και δημιουργείται ένα κλειδί από τα i, j . Ο πίνακας κατακερματισμού που περιέχει τους δρόμους, τροφοδοτείται με το κλειδί αυτό και άρα επιστρέφεται το ArrayList που περιέχει όλους τους δρόμους που ανήκουν στο ίδιο κελί με το σημείο. Στην περίπτωση που στο κελί που βρίσκεται το σημείο δεν υπάρχει κανένας δρόμος, ο αλγόριθμος ψάχνει στα γειτονικά κελιά, προσθέτοντας και αφαιρώντας από το αρχικό κελί και στους δυο άξονες, δηλαδή και οριζόντια και κάθετα, ένα πλήθος κελιών C που καθορίζεται από εμάς και παίρνοντας όλους τους δυνατούς συνδυασμούς κελιών. Η αναζήτηση γίνεται στην τετραγωνική περιοχή που προκύπτει, με κέντρο το αρχικό άδειο κελί, και όπως φαίνεται από το παραπάνω κομμάτι ψευδοκώδικα, το τετράγωνο έχει πλευρά $2C+1$ που μετριέται σε πλήθος κελιών. Αφού προσδιοριστεί ποια από τις δύο περιπτώσεις ισχύει και άρα σε ποιο ή ποια κελιά θα αναζητηθεί ο κοντινότερος δρόμος, εξετάζεται η Ευκλείδεια απόσταση του σημείου από τους δρόμους που ανήκουν σε αυτό ή αυτά και η μικρότερη από αυτές δίνει την σχέση απόστασης nearest. Κατόπιν, επιστρέφεται το id του κοντινότερου δρόμου καθώς και η απόσταση του από το σημείο.

Πρέπει να αναφερθεί, ότι στα πλαίσια της εργασίας αυτής, έχει γίνει η παραδοχή ότι ο κοντινότερος δρόμος για κάθε σημείο, βρίσκεται στο κελί που ανήκει ο δρόμος αυτός. Υπάρχει η περίπτωση αυτό να μην ισχύει και το σημείο να έχει κοντινότερο δρόμο σε γειτονικό κελί. Αυτό μπορεί να συμβεί αν το σημείο βρίσκεται σχετικά κοντά σε κάποια από τις πλευρές-όρια του κελιού και ένας δρόμος βρίσκεται πολύ κοντά στο όριο αυτό από την πλευρά του γειτονικού κελιού. Αυτή η περίπτωση, δύναται να καλυφθεί χρησιμοποιώντας την προσέγγιση που αναφέρθηκε παραπάνω για την περίπτωση του άδειου κελιού.



Εικόνα 27 – Σχέση nearest

Για να βρεθεί ποια σημεία ενδιαφέροντος έχουν κοινά σημεία με τον κύκλο που ορίζουμε κάθε φορά, δηλαδή βρίσκονται εντός του, εξ ολοκλήρου ή μερικώς, ο αλγόριθμος λειτουργεί ως εξής:

```
/*find POIs in each circle*/
while(circles source has more lines){
    createCircle(center, radius);
    get bounding box of circle;
    minX ← bbox.getMinX;
    maxX ← bbox.getMaxX;
    minY ← bbox.getMinY;
    maxY ← bbox.getMaxY;
    (i1,j1) ← findGridCell(minX, minY); //leftDown
    (i2,j2) ← findGridCell(minX, maxY); //leftUp
    (i3,j3) ← findGridCell(maxX, minY); //rightDown
    (i4,j4) ← findGridCell(maxX, maxY); //rightUp
    find minI, maxI, minJ, maxJ;
    for(i=minI; i <= maxI; i++){
        for(j=minJ; j <= maxJ; j++){
            key ← 'ij';
            keyList.add(key);
        }
    }
    for(i=0; i < keyList.length; i++){
```

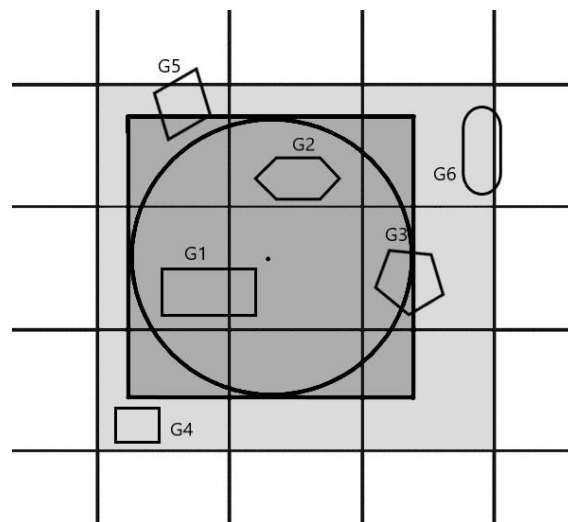
```

polygonList = POIHashTable.get(keyList.get(i));
for(k=0; k < polygonList.length; k++){
    if(polygon intersects with circle){
        return polygon_id;
    }
}
}
}

```

Αρχικά, δημιουργείται ο κύκλος με κέντρο και ακτίνα που έχει οριστεί από εμάς και προσδιορίζεται το τετράγωνο που τον εγγράφει. Στη συνέχεια, βρίσκεται σε ποιο κελί του πλέγματος ανήκει η καθεμία από τις τέσσερις γωνίες του τετραγώνου. Πρέπει όμως να είναι γνωστό και ποια είναι όλα τα ενδιαμέσως κελιά, δηλαδή τα εσωτερικά του τετραγώνου, ώστε να γίνουν συγκρίσεις με τα πολύγωνα που περιέχουν, και τον κύκλο. Για το λόγο αυτόμ πρέπει να βρεθούν τα ελάχιστα και μέγιστα i και j .

Χρησιμοποιώντας όλους τους δυνατούς συνδυασμούς, γίνεται προσπέλαση των κελιών και για κάθε ένα δημιουργείται ένα κλειδί από τα i, j του και προστίθεται σε μια λίστα. Σε αυτό το σημείο, έχει δημιουργηθεί μια λίστα που περιέχει τα κλειδιά όλων των κελιών που βρίσκονται εντός του τετραγώνου που εγγράφει τον κύκλο. Κατόπιν, κάθε κλειδί τροφοδοτείται στον πίνακα κατακερματισμού που περιέχει τα σημεία ενδιαφέροντος και επιστρέφεται το αντιστοιχο ArrayList για το καθένα. Κάθε πολύγωνο που περιέχει το κάθε ArrayList θα συγκριθεί με τον κύκλο και αν έχουν κοινά σημεία, δηλαδή υπάρχει σχέση intersects, επιστρέφεται το id του πολυγώνου-σημείου ενδιαφέροντος.



Εικόνα 28 – Σχέση intersects

Όπως φαίνεται στην παραπάνω εικόνα, όλα τα πολύγωνα είναι εντός των κελιών που βρέθηκε ότι διασχίζει το τετράγωνο που εγγράφει τον κύκλο, όμως ο αλγόριθμος θα επιστρέψει μόνο τα G1, G2 και G3, αφού μόνο αυτά πλοιορούν την σχέση intersects με τον κύκλο.

3.4 Πολυπλοκότητα αλγορίθμων

Έστω ότι στην πηγή των σημείων υπάρχουν m σημεία και στην πηγή των ευθύγραμμων τμημάτων ή των πολυγώνων υπάρχουν n ευθύγραμμα τμήματα ή πολύγωνα (γεωμετρίες). Ο brute force αλγόριθμος θα συγκρίνει πάντα κάθε σημείο με όλα τα ευθύγραμμα τμήματα ή πολύγωνα, άρα θα γίνουν $m \times n$ συγκρίσεις συνολικά. Εάν οι τιμές m, n είναι παρομοίου, πολύ μεγάλου μεγέθους ο αλγόριθμος αυτός παρουσιάζει τετραγωνική πολυπλοκότητα $\Theta(n^2)$.

Ο grid αλγόριθμος διαιρεί την συνολική χωρική περιοχή σε κελιά ίσου μεγέθους. Έστω ότι ο μέσος όρος γεωμετρικών αντικειμένων σε κάθε κελί είναι c , το οποίο είναι προφανώς πολύ μικρότερο από το n , αφού προκύπτει διαιρώντας το n με το πλήθος των κελιών. Για την περίπτωση της σχέσης nearest ο αλγόριθμος συγκρίνει κάθε σημείο με τα ευθύγραμμα τμήματα που ανήκουν στο κελί που ανήκει το ίδιο. Άρα θα γίνουν κατά μέσο όρο $m \times c$ συγκρίσεις. Εάν οι τιμές m, n είναι παρομοίου, πολύ μεγάλου μεγέθους ο αλγόριθμος αυτός παρουσιάζει γραμμική πολυπλοκότητα $\Theta(n)$, αφού το c είναι μικρός αριθμός σε σχέση με το n και άρα δεν παίζει καθοριστικό ρόλο στο πως αυξάνεται το αλγοριθμικό κόστος για πολύ μεγάλες τιμές του n . Όσον αφορά τη σχέση intersects ο αλγόριθμος για κάθε κύκλο (σημείο-ακτίνα), θα αναζητήσει σε όσα κελιά τέμνει ο κύκλος. Ο αριθμός των κελιών αυτών είναι μια συνάρτηση της ακτίνας του κύκλου, $k = f(r)$. Άρα οι συγκρίσεις που θα γίνουν είναι $m \times c \times k$. Όπως και το c , το k είναι πολύ μικρός αριθμός για να παίζει καθοριστικό ρόλο στην αύξηση του αλγοριθμικού κόστους, άρα η πολυπλοκότητα είναι $\Theta(n)$, δηλαδή γραμμική, και σε αυτή την περίπτωση.

Αξίζει να αναφερθεί ότι στην ακραία περίπτωση που το πλήθος κελιών που ορίζεται για τον grid αλγόριθμο είναι 1, άρα όλη η γεωγραφική περιοχή ανήκει σε ένα μόνο κελί, η επίδοση του θα μειωθεί στο ίδιο επίπεδο με του brute force, αφού κάθε σημείο θα χρειαστεί να συγκριθεί με όλες τις γεωμετρίες.

4

Πειραματική μελέτη και υλοποίηση

Στο κεφάλαιο αυτό, θα αναλυθεί ο τύπος και η δομή των πηγών δεδομένων που χρησιμοποιήθηκαν, θα γίνει αναφορά στην γλώσσα προγραμματισμού Java, με την οποία υλοποιήθηκαν οι αλγόριθμοι και στο QGIS, με το οποίο έγινε επαλήθευση των αποτελεσμάτων, και τέλος, θα παρουσιαστεί μια σειρά πειραμάτων και συγκρίσεων των αλγορίθμων.

4.1 Comma-separated values (csv)

Αρχεία .csv είναι αρχεία κειμένου, τα οποία χρησιμοποιούν ένα κόμμα (,) για να διαχωρίσουν τιμές μεταξύ τους. Τα δεδομένα που αποθηκεύουν είναι σε μορφή πίνακα και κάθε γραμμή του αρχείου αντιπροσωπεύει μια εγγραφή. Κάθε εγγραφή αποτελείται από ένα ή περισσότερα πεδία, και κάθε πεδίο διαχωρίζεται από το επόμενο του με ένα κόμμα. Στη χρήση του ως διαχωριστή, οφείλεται η ονομασία του συγκεκριμένου τύπου αρχείων.

Ο τύπος αρχείων αυτός, είναι πάνω από μία δεκαετία προγενέστερος των προσωπικών υπολογιστών (PC) και χρησιμοποιείται ευρέως σε καταναλωτικές, επιχειρησιακές και επιστημονικές εφαρμογές.

Ένα παράδειγμα αρχείου .csv το οποίο περιέχει στοιχεία για την πώληση αυτοκινήτων. Συγκεκριμένα έτος, μήνα, ημέρα και τοποθεσία πώλησης, καθώς και το μοντέλο του οχήματος.

```
year,month,day,location,carmodel
2018,"may",23,"Athens","BMW M3"
2019,"november",2,"Thessaloniki","VW Golf"
2015,"june",16,"Chania","Fiat Punto"
2017,"december",8,"Athens","Toyota Yaris"
```

Εικόνα 29- Παράδειγμα δομής αρχείου csv

πάρκο, στάδιο κλπ και τη γεωμετρία του σε WKT. Όπως έχει αναφερθεί και παραπάνω, τα γεωμετρικά σχήματα που τα αναπαριστούν είναι πολύγωνα.

```
osm_id,code,fclass,geo
"8701114","2256",stadium,"MultiPolygon (((21.750784 38.2217707, 21.75079
"2828","2722",museum,"MultiPolygon (((23.7318334 37.9887141, 23.7318602
"10939654","2732",ruins,"MultiPolygon (((21.7487459 38.2480761, 21.7491
"10973125","2204",park,"MultiPolygon (((23.7342871 37.9720788, 23.73429
"10973127","2204",park,"MultiPolygon (((23.7334316 37.9710865, 23.73348
"10973689","2732",ruins,"MultiPolygon (((23.7261966 37.9715901, 23.7270
"10973689","2721",attraction,"MultiPolygon (((23.7261966 37.9715901, 23.
"10973697","2732",ruins,"MultiPolygon (((23.727098 37.9706378, 23.72712
"10973697","2721",attraction,"MultiPolygon (((23.727098 37.9706378, 23.
"14313699","2256",stadium,"MultiPolygon (((22.8469683 40.6415101, 22.84
"14374667","2015",graveyard,"MultiPolygon (((22.9823651 40.5844063, 22.9
"14379606","2081",university,"MultiPolygon (((22.9538673 40.6315726, 22.
"14616813","2015",graveyard,"MultiPolygon (((23.0562196 40.312265, 23.05
```

Εικόνα 32- Πηγή 3: Σημεία Ενδιαφέροντος

Η τέταρτη και τελευταία πηγή χωρικών δεδομένων περιλαμβάνει συντεταγμένες σημείων που αναπαριστούν κέντρα κύκλων και ακτίνες των κύκλων αυτών σε μέτρα.

```
xC,yC,radius
23.80801195,38.00069973,200
26.3114339,41.19535071,400
23.73879924,37.94563304,170
23.73080404,37.97152932,50
```

Εικόνα 33- Πηγή 4: Κύκλοι

Η δεύτερη και η τέταρτη πηγή δεν περιέχουν WKT, και αυτό γιατί οι απαραίτητες μετατροπές γίνονται εντός του αλγορίθμου.

Οι παραπάνω πηγές χρησιμοποιούνται από τον αλγόριθμο ολοκλήρωσης δεδομένων που αναπτύχθηκε και τα δεδομένα που περιέχουν αλληλεπιδρούν μεταξύ τους καθόλη τη διάρκεια λειτουργίας του, έτσι ώστε να ανακαλυφθούν οι ζητούμενες σχέσεις, μεταξύ των γεωμετρικών αντικειμένων.

4.3 Java και βιβλιοθήκες

Για την ανάπτυξη του αλγορίθμου ολοκλήρωσης χωρικών δεδομένων χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java.

Η Java είναι μια γλώσσα γενικού σκοπού, η οποία αναπτύχθηκε τη δεκαετία του 1990 από την Sun Microsystems. Πριν την δημιουργία της, η C++ ήταν η κυρίαρχη και πιο προτιμώμενη γλώσσα προγραμματισμού. Ο κυριότερος στόχος των δημιουργών της

Java ήταν να υλοποιήσουν μια νέα γλώσσα, η οποία θα εξασφάλιζε τα περισσότερα από αυτά που προσέφερε η C++, ενώ θα αφαιρούσε πολλά περίπλοκα και σπάνια χρησιμοποιούμενα στοιχεία. Το εγχείρημα αυτό, γνώρισε μεγάλη επιτυχία καθώς η Java σήμερα είναι από τις πιο δημοφιλείς γλώσσες προγραμματισμού στον κόσμο, με πάνω από 9 εκατομμύρια developers.

Η Java έχει τα εξής βασικά χαρακτηριστικά:

- Είναι αντικειμενοστραφής. Αυτό σημαίνει ότι το λογισμικό οργανώνεται ως ένας συνδυασμός διαφορετικών ειδών αντικειμένων που ενσωματώνουν δεδομένα και συμπεριφορά.
- Είναι ανεξάρτητη από την πλατφόρμα υλοποίησης. Ο κώδικας java μπορεί να τρέξει σε όλες τις υποστηριζόμενες πλατφόρμες όπως Windows, Linux, MacOS κλπ ανεξάρτητα από διαφορές στην αρχιτεκτονική. Το χαρακτηριστικό αυτό έχει πάρει την ονομασία WORA (Write Once and Run Anywhere)
- Είναι ασφαλής. Η Java είναι γνωστή για το υψηλό επίπεδο ασφάλειας που προσφέρει. Αυτό οφείλεται κυρίως στο γεγονός ότι οι εφαρμογές java τρέχουν μέσα σε μια εικονική μηχανή και δεν έχουν άμεση επαφή με το λειτουργικό σύστημα.

Φυσικά, η Java έχει πολλά περισσότερα χαρακτηριστικά, όμως η αναφορά και η ανάλυση τους είναι πέρα από το αντικείμενο της εργασίας αυτής.

Στην επιστήμη των υπολογιστών και συγκεκριμένα στην ανάπτυξη προγραμματιστικών εφαρμογών, εμφανίζονται μοτίβα, δηλαδή χρειάζεται να επαναληφθούν συγκεκριμένες ενέργειες πολλές φορές. Αυτό το φαινόμενο καθιστά αναγκαία την δημιουργία μεθόδων, δηλαδή μικρών κομματιών κώδικα, τα οποία φτιάχνονται μία φορά και καλούνται όποτε πρέπει να συμβεί η ενέργεια για την οποία προορίζονται. Σε μεγαλύτερη κλίμακα, συλλογές μεθόδων οι οποίες εξυπηρετούν έναν κοινό σκοπό ονομάζονται βιβλιοθήκες.

Οι βιβλιοθήκες που χρησιμοποιήθηκαν κατά την ανάπτυξη του αλγορίθμου ολοκλήρωσης χωρικών δεδομένων, που είναι αντικείμενο της εργασίας αυτής, είναι η Geotools και η OpenCSV.

Η Geotools είναι μια βιβλιοθήκη η οποία παρέχει δυνατότητες διαχείρισης χωρικών και γεωγραφικών δεδομένων. Δημιουργήθηκε το 1998 και είχε στόχο να καταστήσει δυνατή τη δημιουργία διαδραστικών εφαρμογών οπτικοποίησης γεωγραφικών δεδομένων.

Η OpenCSV είναι μια βιβλιοθήκη η οποία περιέχει εργαλεία για την δημιουργία java αντικειμένων από εγγραφές αρχείων csv (Comma-separated values). Χρησιμοποιήθηκε ώστε η κάθε γραμμή των πηγών δεδομένων που περιγράφηκαν παραπάνω να μετατραπεί σε ένα γεωμετρικό αντικείμενο κατά την λειτουργία του αλγορίθμου που αναπτύχθηκε.

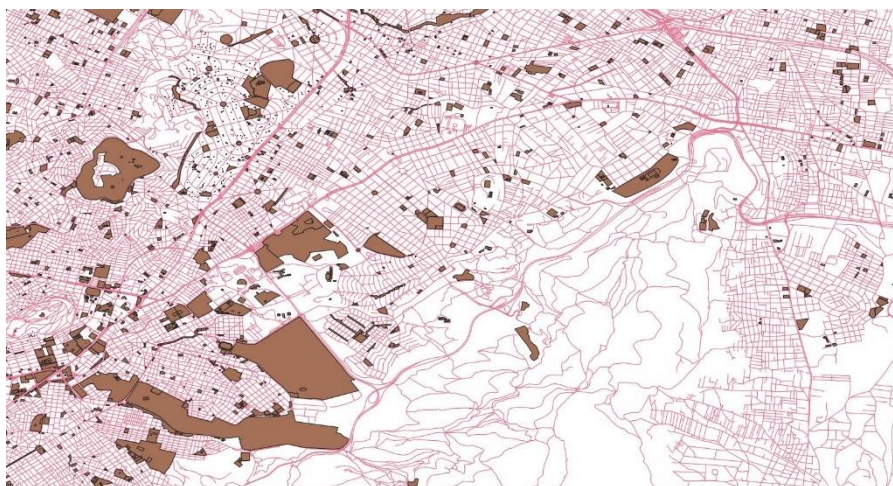
4.4 QGIS

Το QGIS είναι ένα γεωγραφικό πληροφοριακό σύστημα (GIS) το οποίο χρησιμοποιήθηκε για την οπτικοποίηση των χωρικών δεδομένων με τα οποία τροφοδοτήθηκε ο αλγόριθμος που αναπτύχθηκε. Έπαιξε σημαντικό ρόλο στην επαλήθευση των αποτελεσμάτων του αλγορίθμου καθώς και στην δημιουργία γεοχωρικών δεδομένων τα οποία θα χρησιμοποιούνταν για διαφόρων ειδών δοκιμές.



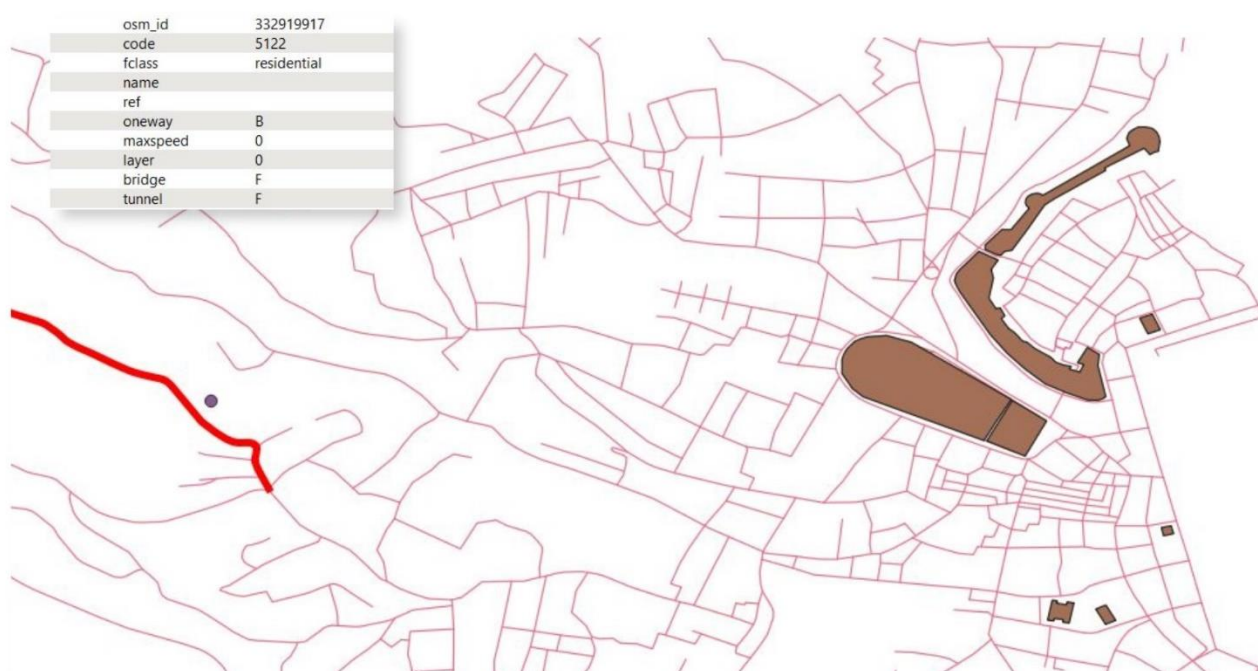
Εικόνα 34 – Απεικόνιση της Ελλάδας στο QGIS

Τα δεδομένα που επιλέχθηκαν για την τροφοδότηση του αλγορίθμου αναπαριστούν το οδικό δίκτυο της Ελλάδας και σημεία ενδιαφέροντος πάνω σε αυτό, όπως φαίνεται στην παραπάνω εικόνα.



Εικόνα 35 – QGIS μεγέθυνση

Μεγεθυνόνοντας, φαίνεται με μεγαλύτερη λεπτομέρεια η οπτική αναπαράσταση των δεδομένων που αντιπροσωπεύουν το οδικό δίκτυο και τα σημεία ενδιαφέροντος.



Εικόνα 36 – QGIS id κοντινότερου δρόμου

Δημιουργώντας ένα σημείο δίπλα σε οποιονδήποτε δρόμο, φαίνεται το id του δρόμου αυτού, που είναι ο κοντινότερος, και μπορεί να διασταυρωθεί εάν ο αλγόριθμος επιστρέφει ορθά αποτελέσματα.



Εικόνα 37 – QGIS id σημείων ενδιαφέροντος

Ομοίως, μπορούν να ελεγχθούν τα id των σημείων ενδιαφέροντος εντός μιας εμβέλειας από κάποιο σημείο που δημιουργήσαμε, το οποίο θα αποτελέσει το κέντρο ενός κύκλου, και να επαληθευτεί ότι τα id που επιστρέφει ο αλγόριθμος είναι αυτά που πρέπει.

4.5 Πειραματική μελέτη

Μετά το πέρας της υλοποίησης του αλγορίθμου, πραγματοποιήθηκαν ορισμένα πειράματα. Αυτά περιλαμβάνουν μετρήσεις χρόνων εκτέλεσης διαφοροποιώντας συγκεκριμένες παραμέτρους, καταγραφή των μετρήσεων και σύγκριση τους ώστε να εξαχθούν συμπεράσματα για την συμπεριφορά και την επίδοση του αλγορίθμου υπό διαφορετικά σενάρια, καθώς και σε σχέση με έναν απλούστερο αλγόριθμο ολοκλήρωσης χωρικών δεδομένων. Οι μετρήσεις αναλύονται παρακάτω και παρίστανται διαγραμματικά.

4.5.1 Επίδοση αλγορίθμου για διαφορετικά πλήθη κελιών πλέγματος

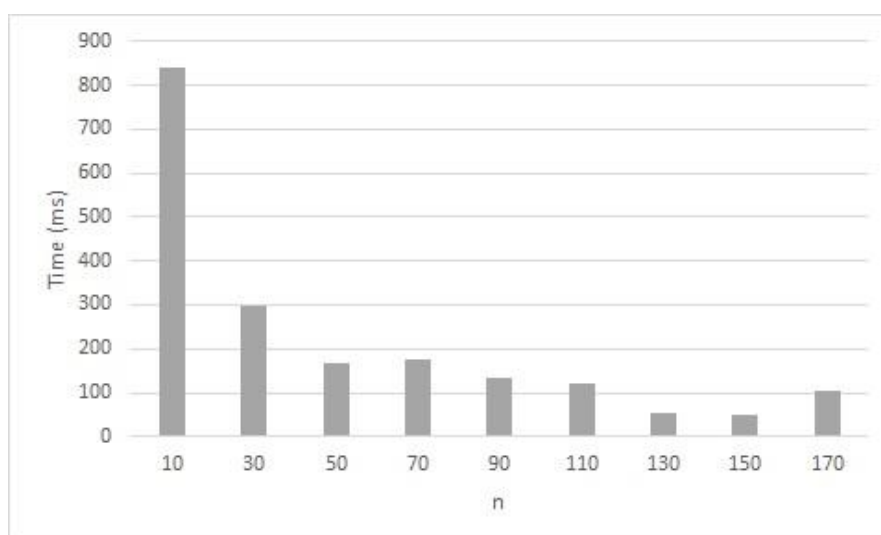
Σκοπός του πειράματος αυτού είναι να αξιολογηθεί η επίδοση του grid αλγορίθμου για διάφορα πλήθη κελιών του πλέγματος και για τις δύο σχέσεις που αναζητά και να βρεθεί το βέλτιστο πλήθος, αυτό δηλαδή για το οποίο επιτυγχάνεται η καλύτερη επίδοση.

Όπως έχει αναφερθεί στην αντίστοιχη ενότητα το πλέγμα που χρησιμοποιεί ο αλγόριθμος αποτελείται από $n \times n$ κελιά. Στο πείραμα αυτό μεταβάλλεται η

παράμετρος n , ξεκινώντας απο μία τιμή και αυξάνοντας την σταδιακά. Σε καθεμία περίπτωση μετριέται ο χρόνος εκτέλεσης του αλγορίθμου, δηλαδή πόσο χρόνο χρειάζεται για να ανακαλύψει τις ζητούμενες σχέσεις μεταξύ των γεωμετρικών αντικειμένων που προκύπτουν απο τα χωρικά δεδομένα με τα οποία έχει τροφοδοτηθεί.

Αρχικά, αξιολογήθηκε η επίδοση για την εύρεση κοντινότερου δρόμου για κάθε σημείο στα δεδομένα μας (σχέση nearest). Για κάθε n μετρήθηκε ο χρόνος εκτέλεσης του αλγορίθμου για 5 σημεία και από αυτά υπολογίστηκε ο μέσος χρόνος εκτέλεσης.

n	10	30	50	70	90	110	130	150	170
Time	839,03	297,76	168,06	174,42	134,12	121,88	52,44	48,01	104,46



Εικόνα 38 – Απόδοση grid αλγορίθμου για τη σχέση nearest

Παραπάνω βλέπουμε, οτι το πείραμα ξεκινά με την τιμή $n=10$, δηλαδή με το πλέγμα να αποτελείται απο 100 κελιά, αυξάνοντας το n κατά 20 μονάδες τη φορά. Η αύξηση του n συνεχίζεται, σταματώντας στο σημείο που η βελτίωση παύει και ο χρόνος εκτέλεσης αρχίζει να αυξάνεται. Παρατηρείται οτι ο χρόνος εκτέλεσης του αλγορίθμου μειώνεται σταδιακά μέχρι που φτάνει την βέλτιστη μέση τιμή του, που είναι 48ms ανά σημείο, για $n=150$. Από αυτό το σημείο και μετά, ο χρόνος αρχίζει να αυξάνεται ξανά.

Στην συνέχεια αξιολογήθηκε η επίδοση του αλγορίθμου για την εύρεση όλων των σημείων ενδιαφέροντος εντός ενός κύκλου. Για κάθε n μετρήθηκε ο χρόνος εκτέλεσης του αλγορίθμου για 10 κέντρα και από αυτά υπολογίστηκε ο μέσος χρόνος

εκτέλεσης. Το πείραμα αυτό πραγματοποιήθηκε σε δύο εκδοχές. Μία με μεταβαλλόμενη ακτίνα με βάση το μέγεθος των κελιών του πλέγματος και μία με την ακτίνα των κύκλων σταθερή. Αξίζει να αναφερθεί ότι ενώ υπάρχει περίπτωση κάθε κελί του πλέγματος να έχει όλες τις πλευρές του ίσες, στις περισσότερες περιπτώσεις αυτό δεν συμβαίνει, αφού το γεωγραφικό κομμάτι που καλύπτει το πλέγμα είναι κάποιας μορφής ορθογώνιο και όχι τετράγωνο. Αυτό σημαίνει ότι και οι διαφορές μέγιστης-ελάχιστης τετμημένης και μέγιστης-ελάχιστης τεταγμένης θα είναι διαφορετικές και άρα διαιρούμενες με το n θα παράγουν διαφορετικά StepX και StepY.

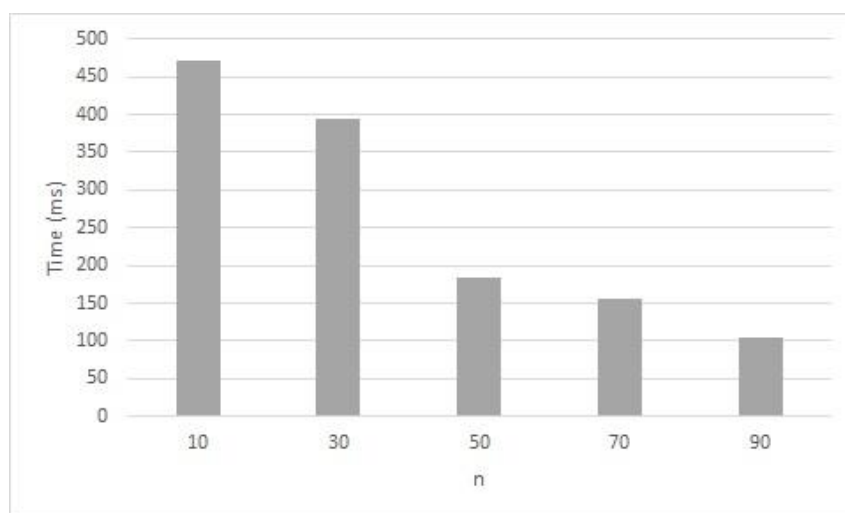
Στην πρώτη περίπτωση που αναφέρθηκε παραπάνω, η μεταβλητή ακτίνα υπολογίστηκε με τον ακόλουθο τρόπο:

$$averageStep = \frac{StepX + StepY}{2}$$

$$r = averageStep * 0.25$$

Η ακτίνα είναι το ένα τέταρτο του μέσου όρου των πλευρών κάθε κελιού. Όσο το n , δηλαδή και το πλήθος των κελιών αυξάνεται, τόσο μειώνεται το μέγεθος τους και άρα και η ακτίνα των κύκλων. Παρακάτω φαίνονται τα διαγράμματα με τους αντίστοιχους μέσους χρόνους εκτέλεσης.

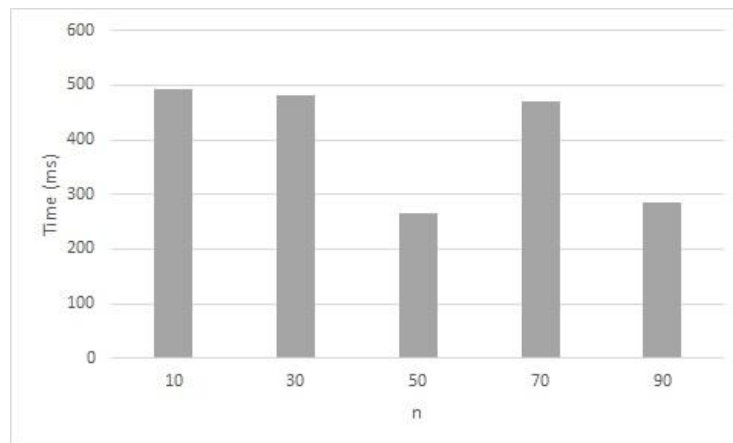
n	10	30	50	70	90
Time	471,31	394,85	184,54	155,75	105,2



Εικόνα 39 – Απόδοση grid αλγορίθμου για τη σχέση intersects με μεταβλητή ακτίνα

Όπως είναι λογικό οι χρόνοι εκτέλεσης πέφτουν όσο μικραίνει η ακτίνα, όμως για να αξιολογηθεί η επίδοση του αλγορίθμου θα χρησιμοποιηθεί ως, σταθερή πλέον ακτίνα ο μέσος όρος των ακτινών των παραπάνω περιπτώσεων και το n θα μεταβληθεί όπως πριν.

n	10	30	50	70	90
Time	493,95	480,03	266,21	470,24	285,65



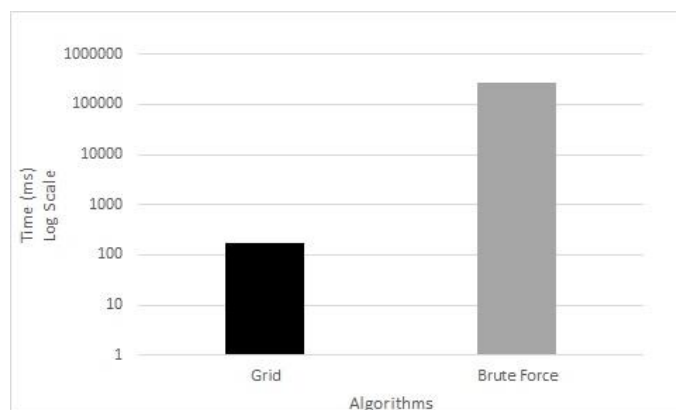
Εικόνα 40 – Απόδοση grid αλγορίθμου για τη σχέση intersects με σταθερή ακτίνα

Παρατηρείται ότι ο καλύτερος χρόνος εκτέλεσης είναι 266ms ανα κύκλο και επιτυγχάνεται για $n=50$. Από εκεί και μετά ο χρόνος αρχίζει και αυξάνεται.

4.5.2 Σύγκριση αλγορίθμων grid – brute force

Σκοπός του πειράματος αυτού είναι, αφού έχει προσδιοριστεί η βέλτιστη τιμή του n για τον grid αλγόριθμο, να συγκριθεί με τον brute force αλγόριθμο και να προσδιοριστεί η διαφορά των επιδόσεων τους και για τις δύο σχέσεις.

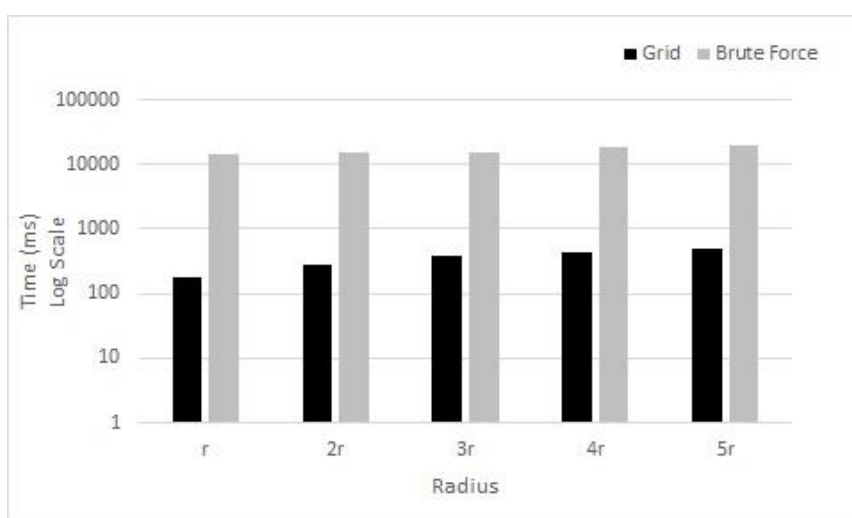
Για το πείραμα αυτό τέθηκε η μεσαία τιμή $n=50$ για τα κελιά του πλέγματος και κρατώντας τη σταθερή συγκρίθηκε η επίδοση του grid αλγορίθμου με την επίδοση του brute force αλγορίθμου, ο οποίος για να βρει τις απαιτούμενες σχέσεις μεταξύ των γεωμετρικών αντικειμένων συγκρίνει όλα τα σημεία με όλους τους δρόμους και όλους τους κύκλους με όλα τα σημεία ενδιαφέροντος.



Εικόνα 41 – Σύγκριση αλγορίθμων grid-brute force για την σχέση nearest

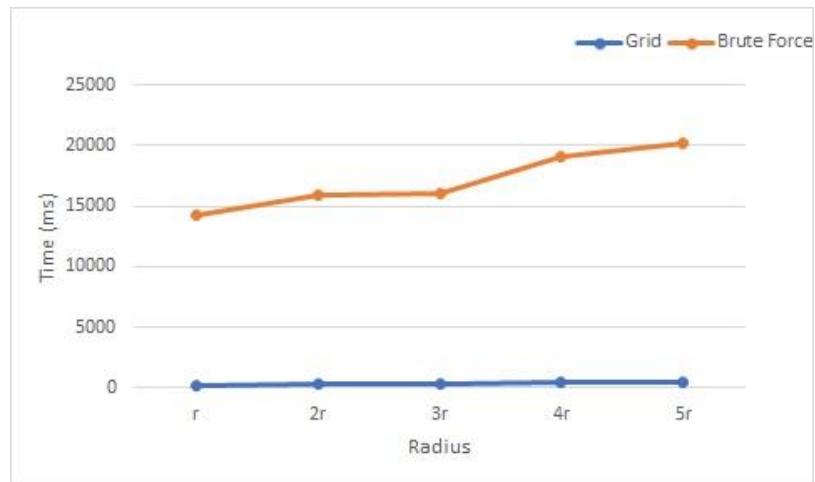
Όσον αφορά τη σχέση nearest για σημεία-δρόμους, η συγκριτική αξιολόγηση απεικονίζεται στο παραπάνω διάγραμμα σε λογαριθμική κλίμακα, όπου παρουσιάζεται ο μέσος χρόνος που χρειάζεται ο κάθε αλγόριθμος.

Για την σχέση intersects για κύκλους-σημεία ενδιαφέροντος, το πείραμα ξεκινά με την μέση ακτίνα που υπολογίστηκε στο προηγούμενο πείραμα και αρχίζοντας από το 1, σε κάθε βήμα αυξάνεται πολλαπλασιαζόμενη με τον επόμενο μεγαλύτερο ακέραιο.



Εικόνα 42 – Σύγκριση αλγορίθμων grid-brute force για την σχέση intersects

Παρατηρείται ότι και οι δύο αλγόριθμοι παρουσιάζουν σχετικά μικρή αύξηση στον χρόνο εκτέλεσης τους αλλά η διαφορά μεταξύ των δύο παραμένει εξαιρετικά μεγάλη για όλα τα μεγέθη ακτινών. Αυτό απεικονίζεται και στο παρακάτω διάγραμμα σε μη λογαριθμική κλίμακα.



Εικόνα 43 – Σύγκριση αλγορίθμων grid-brute force για την σχέση intersects σε line chart

5

Συμπεράσματα και μελλοντικές επεκτάσεις

Στην εργασία αυτή, υλοποιήθηκε ένας αλγόριθμος ολοκλήρωσης χωρικών δεδομένων που έχει στόχο την εύρεση σχέσεων μεταξύ γεωμετρικών αντικειμένων, παράγοντας έτσι νέα, χρήσιμη πληροφορία. Ο στόχος αυτός επιτεύχθηκε με αποδοτικό τρόπο, χωρίζοντας το γεωγραφικό κομμάτι που αναπαριστούν τα χωρικά δεδομένα σε πολλά μικρότερα με τη χρήση πλέγματος και την αποθήκευση των αντικειμένων σε πίνακες κατακερματισμού, ώστε να μειωθούν σε μεγάλο βαθμό οι απαραίτητες συγκρίσεις. Υλοποιήθηκε επίσης ένας δεύτερος απλός αλγόριθμος, ο οποίος δεν χρησιμοποιεί τις προαναφερθείσες προσεγγίσεις. Κάνοντας συγκρίσεις χρόνων εκτέλεσης των δύο αλγορίθμων οι οποίες παρουσιάστηκαν διαγραμματικά, δείξαμε ότι ο πρώτος είναι δεκάδες με εκατοντάδες φορές γρηγορότερος, ανάλογα την σχέση.

Συγκεκριμένα, χρησιμοποιήθηκαν δύο σύνολα χωρικών δεδομένων για την τροφοδότηση των αλγορίθμων και τις απαραίτητες δοκιμές λειτουργίας τους, τα οποία αντλήθηκαν από το OpenStreetMap. Το πρώτο αναπαριστά το οδικό δίκτυο της Ελλάδας και το δεύτερο σημεία ενδιαφέροντος εντός του Ελλαδικού χώρου.

Τέλος, αναλύθηκαν λεπτομερώς όλα τα στάδια υλοποίησης του συστήματος καθώς και το θεωρητικό υπόβαθρο που είναι αναγκαίο για την κατανόηση τους.

Όπως έχει αναφερθεί στην αντίστοιχη ενότητα, όλα τα κελιά του πλέγματος έχουν ίσο μέγεθος μεταξύ τους. Υπάρχει δυνατότητα να επιτευχθεί καλύτερη επίδοση του αλγορίθμου αν με κάποιο τρόπο προσδιορίζεται η πυκνότητα του των χωρικών δεδομένων σε διάφορα μέρη της συνολικής περιοχής και μικραίνοντας το μέγεθος των κελιών στις πυκνές περιοχές, αφού εκεί θα γίνουν περισσότερες συγκρίσεις.

Επίσης, στα πλαίσια της εργασίας αυτής χρησιμοποιήθηκαν ως στόχος των αλγορίθμων, η τοπολογική σχέση intersects και η σχέση απόστασης nearest. Ο

κώδικας δύναται να επεκταθεί ώστε να ανακαλύπτει περισσότερες σχέσεις μεταξύ των γεωμετρικών αντικειμένων.

Τέλος, θα μπορούσαν να υπολογίζονται δεδομένα για κάθε κελί του πλέγματος, όπως πλήθος γεωμετριών, πυκνότητα και ποσοστά διαφορετικών ειδών γεωμετριών και στη συνέχεια να εξάγονται στατιστικά συμπεράσματα όπως μέση τιμή και διακύμανση για το αρχικό σύνολο χωρικών δεδομένων.

Βιβλιογραφία

- [1] **A. S. Fotheringham, c. Brunson, and m. Charlton:** Quantitative geography: Perspectives on spatial data analysis, Mikelbank, B.A. Geogr. Anal. 2001, 33, 370–372
- [2] **AnHai Doan, Alon Y. Halevy, Zachary G. Ives:** Principles of Data Integration. Morgan Kaufmann 2012, ISBN 978-0-12-416044-6, pp. I-XVIII, 1-497
- [3] **Michael R. Genesereth:** Data Integration: The Relational Logic Approach. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers 2010
- [4] **Dinesh P. Mehta, Sartaj Sahni:** Handbook of Data Structures and Applications. Chapman and Hall/CRC 2004, ISBN 978-1-58488-435-4
- [5] **Michael T. Goodrich, Roberto Tamassia:** Data structures and algorithms in Java. World wide series in computer science, Wiley 1998, ISBN 978-0-471-19308-1, pp. I-XVIII, 1-738
- [6] **Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein:** Introduction to Algorithms, Second Edition. The MIT Press and McGraw-Hill Book Company 2001, ISBN 0-262-03293-7
- [7] **Jens Bleiholder, Felix Naumann:** Data fusion. ACM Comput. Surv. 41(1): 1:1-1:41 (2008)
- [8] **Spatial Relations:** https://en.wikipedia.org/wiki/Spatial_relation
- [9] **Brent Hall, Michael G. Leahy:** Open Source Approaches in Spatial Data Handling, Springer Publishing Company Incorporated 2008, ISBN:354074830X 9783540748304
- [10] **Menno-Jan Kraak ,Ferjan Ormeling:** Cartography: visualization of spatial data (3rd ed.), London, Pearson Education Ltd, 2010
- [11] **Marco Painho, Maribel Yasmina Santos, Hardy Pundt:** Geospatial Thinking - International AGILE'2010 Conference, Guimarães, Portugal, 11-14 May 2010. Lecture Notes in Geoinformation and Cartography, Springer 2010, ISBN 978-3-642-12325-2