ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Ευρετηρίαση σε Ερωτήματα k Πλησιέστερων Γειτόνων στην Κύρια Μνήμη

Διπλωματική Εργασία

Αχιλλέας Μιχαλόπουλος

Επιβλέπων: Μιχαήλ Βασιλακόπουλος

Σεπτέμβριος 2022

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Ευρετηρίαση σε Ερωτήματα k Πλησιέστερων Γειτόνων στην Κύρια Μνήμη

Διπλωματική Εργασία

Αχιλλέας Μιχαλόπουλος

Επιβλέπων: Μιχαήλ Βασιλακόπουλος

Σεπτέμβριος 2022

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Indexing on k Nearest-Neighbors Queries in Main Memory

Diploma Thesis

Achilleas Michalopoulos

Supervisor: Michael Vassilakopoulos

September 2022

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων **Μιχαήλ Βασιλακόπουλος**
Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Χρήστος Δ. Αντωνόπουλος**
Αναπληρωτής καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Χαρίκλεια Τσαλαπάτα**
Μέλος Ε.ΔΙ.Π., Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Πρώτα απ'όλα θα ήθελα να ευχαριστήσω τους καθηγητές της τριμελής επιτροπής της διπλωματικής μου εργασίας Μιχαήλ Βασιλακόπουλο, Χρήστο Δ. Αντωνόπουλο και Χαρίκλεια Τσαλαπάτα, για τις γνώσεις και την καθοδήγηση που μου παρείχαν καθ'όλη τη διάρκεια της φοίτησής μου στο Τμήμα.

Εν συνεχεία, θα ήθελα να ευχαριστήσω τον καθηγητή Νίκολαο Μαμουλή και τον διδακτορικό φοιτητή Δημήτριο Τσιτσίγκο, του Τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής του Πανεπιστημίου Ιωαννίνων, για την επιλογή του θέματος της εργασίας, καθώς και για την άριστη συνεργασία που είχαμε όλους αυτούς τους μήνες.

Τέλος, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην οικογένειά μου και στους φίλους μου, για την υποστήριξή τους και τις στιγμές που μοιραστήκαμε όλα αυτά τα χρόνια.

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

«Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Δηλώνω επίσης ότι τα αποτελέσματα της εργασίας δεν έχουν χρησιμοποιηθεί για την απόκτηση άλλου πτυχίου. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής».

Ο Δηλών

Αχιλλέας Μιχαλόπουλος

Διπλωματική Εργασία

Ευρετηρίαση σε Ερωτήματα k Πλησιέστερων Γειτόνων στην Κύρια Μνήμη

Αχιλλέας Μιχαλόπουλος

Περίληψη

Στις μέρες μας, παρότι υπάρχουν αρκετές μέθοδοι, οι οποίες διαχειρίζονται αποτελεσματικά τη συλλογή και οργάνωση χωρικών δεδομένων, εξακολουθεί να αποτελεί ζητούμενο, η αναζήτηση νέων προσεγγίσεων, που ενισχύουν ακόμη περισσότερο την αποδοτικότητα των χωρικών ερωτημάτων. Βασισμένοι σε αυτό, προτείνουμε μία νέα μέθοδο, για την ευρετηρίαση αντικειμένων με γεωμετρικές εκτάσεις, στο επίπεδο της Κύριας Μνήμης, η οποία στηρίζεται στη λογική διαμέρισης του χώρου, ως πλέγματος ανεξάρτητων κελιών. Επεκτείνουμε τη συγκεκριμένη διαμέριση, εισάγοντας ένα επιπλέον επίπεδο, με το οποίο διαχωρίζουμε περαιτέρω το περιεχόμενο του κάθε κελιού, σε δεκαέξι κλάσεις. Με αυτό τον τρόπο, εξασφαλίζουμε την αποφυγή διπλότυπων εμφανίσεων στα αποτελέσματα των ερωτημάτων, φαινόμενο που συναντάται διαρκώς στην κατηγορία αυτών των μεθόδων. Εν συνεχεία, αναπτύσσουμε έναν αλγόριθμο για τη διεκπεραίωση ερωτημάτων k πλησιέστερων γειτόνων, ο οποίος συνδέεται άμεσα με τον τρόπο που αποθηκεύουμε τα αντικείμενα στη μνήμη. Τέλος, αξιολογούμε την προσέγγιση μας στο σύνολό της, έναντι δύο παραλλαγών του R-tree, εφαρμόζοντας πειράματα, που ανταποκρίνονται σε πραγματικά σύνολα δεδομένων. Τα αποτελέσματα επιβεβαιώνουν την υπεροχή μας τόσο στο στάδιο της καταγραφής, όσο και στο στάδιο της εκτέλεσης. Πιο αναλυτικά, για κάθε σύνολο αντικειμένων, πετυχαίνουμε χαμηλότερους μέσους χρόνους ευρετηρίασης, έναντι ακόμα της καλύτερης αντίπαλης προσέγγισης που παρουσιάζουμε, καθώς και χαμηλότερους μέσους χρόνους διεκπεραίωσης ερωτημάτων, με σταθερά αυξανόμενη επιτάχυνση, καθώς προχωράμε σε δοκιμές μεγαλύτερων k τιμών.

Λέξεις-κλειδιά:

Χωρική Ευρετηρίαση, Διεκπεραίωση Ερωτημάτων, Κύρια Μνήμη, Χωρικές Βάσεις Δεδομένων, Διαχείριση Δεδομένων

Diploma Thesis

Indexing on k Nearest-Neighbors Queries in Main Memory

Achilleas Michalopoulos

Abstract

Nowadays, although several methods effectively manage the collection and organization of spatial data, it is still a challenge to find new approaches to enhance the performance of spatial queries. Based on this, we propose a new method, for indexing objects with geometric extents, at the Main Memory, which is based on the logic of partitioning the space as a grid of disjoint cells. We extend this partitioning by introducing an additional layer, which divides up further the content of each cell, into sixteen classes. In this way, we ensure that duplicate appearances in the query results are avoided, a phenomenon that is constantly encountered in the class of these methods. Next, we develop an algorithm for processing k Nearest-Neighbors queries, which is related directly to the way we store the objects in memory. Finally, we evaluate our approach in its entirety against two variants of the R-tree by applying experiments corresponding to real datasets. The results confirm our superiority in both the storage and execution stages. More specifically, for each set of objects, we achieve lower mean indexing times, compared to even the best competing approach we present, as well as lower mean query execution times, with steadily increasing speedup as we proceed to test greater k values.

Keywords:

Spatial Indexing, Query Processing, Main Memory, Spatial Databases, Data Management

Πίνακας περιεχομένων

Κατάλογος σχημάτων

Κατάλογος πινάκων

Συντομογραφίες

δηλ.	δηλαδή
π.χ	παραδείγματος χάριν
βλπ	βλέπε
κ.λπ.	και λοιπά
κ.ο.κ	και ούτω καθεξής
ΒΔ	Βάση Δεδομένων
ΣΔΒΔ	Συστήματα Διαχείρισης Βάσεων Δεδομένων
ΣΓΠ	Συστήματα Γεωγραφικών Πληροφοριών
MBR	Minimum Bounding Rectangle
SOP	Space-oriented Partitioning
DOP	Data-oriented Partitioning

Κεφάλαιο 1

Εισαγωγή

Τα Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ), έως και σήμερα, χαρακτηρίζονται ως ο αποτελεσματικότερος τρόπος διαχείρισης δεδομένων. Ως ΣΔΒΔ εννοείται ένα λογισμικό που έχει δύο βασικές λειτουργικότητες: (i) φροντίζει για τη δημιουργία, συντήρηση και επεξεργασία των δεδομένων που αποθηκεύονται σε μία Βάση Δεδομένων (ΒΔ) και (ii) διεκπεραιώνει τα αιτήματα ανάκτησης πληροφοριών των χρηστών, απαλλάσσοντάς τους από οποιαδήποτε γνώση σχετικά με του που/πως είναι αποθηκευμένα τα δεδομένα. Η πρώτη εμφάνιση τους έγινε την περίοδο που η εν λόγω διαχείριση είχε ανατεθεί στα Συστήματα Αρχείων. Ωστόσο δεν άργησε να έρθει η επικράτησή τους, από τη στιγμή που ήταν σε θέση να αντιμετωπίσουν βασικούς περιορισμούς της μέχρι τότε εποχής:

- **Αδυναμία προτυποποίησης**
- **Πλεονασμός και ασυνέπεια δεδομένων**
- **Αδυναμία γρήγορης ανάκαμψης σε πιθανό πρόβλημα**
- **Αδυναμία μερισμού δεδομένων και έλλειψη ελέγχων συγχρονισμού**

Σήμερα μπορεί να παρατηρήσει κανείς ότι τα ΣΔΒΔ έχουν συνδεθεί με όλες τις δραστηριότητες της καθημερινότητάς μας. Σε εφαρμογές όπως οι τραπεζικές συναλλαγές και οι κρατήσεις θέσεων-εισιτηρίων, στις οποίες συλλέγονται δεδομένα απλού τύπου (π.χ. νούμερα, αλφαριθμητικά, ημερομηνίες), η άμεση ανταπόκρισή τους πιστώνεται στο συνδυασμό οργάνωσης, αξιοπιστίας, προστασίας και ταχύτητας που παρέχεται από αυτά τα συστήματα. Βέβαια, σε πιο σύγχρονες εφαρμογές (π.χ. πολυμεσικές), στις οποίες συναντώνται δεδομένα

σύνθετου τύπου, η διαχείριση εξακολουθεί να αποτελεί πρόκληση και θέμα ερευνητικού ενδιαφέροντος. Στην κατηγορία των σύνθετων δεδομένων συγκαταλέγονται και τα χωρικά δεδομένα, τα οποία θα μας απασχολήσουν στην παρούσα εργασία. Στις ενότητες που ακολουθούν εξηγούμε τους τύπους χωρικών δεδομένων, καθώς και τους τρόπους επίτευξης της διαχείρισής τους.

1.1 Διαχείριση χωρικών δεδομένων

Εδώ και δεκαετίες, έχει παρατηρηθεί ότι κατά τη συλλογή και ανάλυση δεδομένων, εμπεριέχεται πληροφορία σχετική με το χώρο [3]. Έτσι, από νωρίς η ερευνητική κοινότητα στράφηκε στην αναζήτηση μεθόδων αξιοποίησής της. Η αρχή έγινε με τα πρώτα Συστήματα Γεωγραφικών Πληροφοριών (ΣΓΠ) [6], τα οποία, χρησιμοποιώντας τα γεωγραφικά χαρακτηριστικά των δεδομένων, κατόρθωσαν να προσφέρουν την αναπαράστασή τους πάνω σε χάρτες. Μετέπειτα η αναπαράσταση αυτή, αποτέλεσε το έναυσμα, ώστε οι χρήστες να θέλουν να υποβάλλουν πιο σύνθετα αιτήματα.

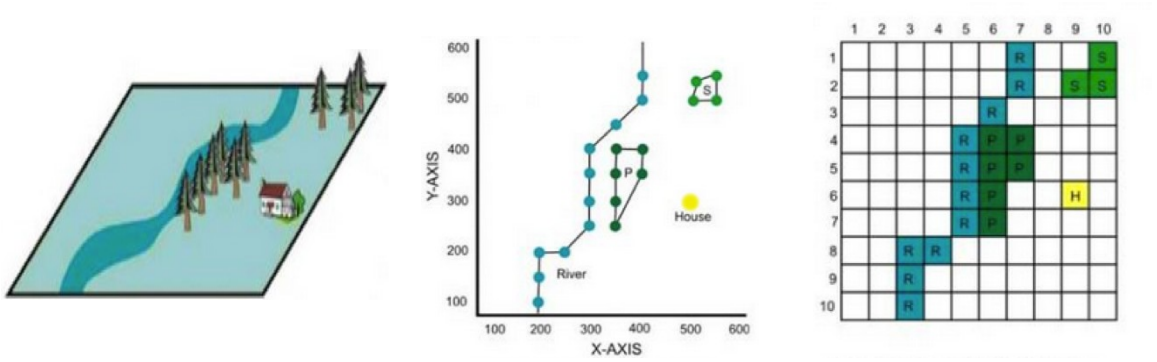
Πλέον, οι τεχνικοί των μοντέρνων ΣΔΒΔ έχουν αρχίσει να ασχολούνται με το κομμάτι των απαραίτητων επεκτάσεων που απαιτούνται στα συστήματά τους, ώστε να είναι εφικτή η διαχείριση χωρικών δεδομένων, από τη στιγμή που το σχεσιακό μοντέλο δε θεωρείται επαρκές για τη διεκπεραίωση ερωτημάτων χωρικού τύπου. Για παράδειγμα, ας υποθέσουμε ότι έχουμε στη διάθεσή μας δεδομένα που αφορούν όλους τους χώρους εστίασης μίας πόλης. Τα ΣΔΒΔ, μέχρι στιγμής, είναι ικανά να ανταποκριθούν σε αιτήματα αναζήτησης όπως το όνομα ενός χώρου ή τον τύπου του (π.χ. εστιατόριο, καφετέρια), αλλά δεν είναι σε θέση να αναζητήσουν τους πλησιέστερους χώρους, βάσει τη τοποθεσία μας.

1.1.1 Τύποι χωρικών δεδομένων

Ο πιο διαδεδομένος τύπος χωρικών δεδομένων που χρησιμοποιούμε κατά την περιγραφή ενός αντικειμένου, είναι η θέση του. Πρόκειται για πληροφορία, η οποία εκφράζεται μέσω ενός συστήματος συντεταγμένων (ανάλογα με το αν το αντικείμενο είναι δισδιάστατο ή τρισδιάστατο), έχοντας όμως τη δυνατότητα να αποθηκευτεί σε μορφή χαμηλότερης διάστασης μετά από την κατάλληλη επεξεργασία της. Σε ορισμένες περιπτώσεις, εκτός από τη θέση του αντικειμένου, μας απασχολεί και η γεωμετρική του έκταση. Ανάλογα με τη κλίμακα του υποβληθέντος αιτήματος, αποφασίζεται αν είναι αναγκαία ή όχι, η αναφορά στην έκταση ενός

αντικειμένου. Στο παράδειγμα με τους χώρους εστίασης, αν η αναζήτηση γίνεται σε όλη την γεωγραφική επιφάνεια μίας πόλης, η χρήση μόνο της θέσης επαρκεί για τους σκοπούς μας. Αν όμως επικεντρωνόμαστε σε μία συγκεκριμένη περιοχή, η αξιοποίηση των εκτάσεων που καταλαμβάνουν οι χώροι, μπορεί να είναι καθοριστική για την ορθότητα του αποτελέσματος.

Για την αναπαράσταση των χωρικών αντικειμένων, ακολουθούμε συγκεκριμένα πρότυπα. Όταν θέλουμε να αναπαραστήσουμε ένα αντικείμενο με βάση τη θέση του, χρησιμοποιούμε ένα απλό σημείο, ενώ αν χρήζει αναγκαία η απεικόνιση της έκτασής του, κάνουμε χρήση σχημάτων (π.χ. πολυγραμμές για την απεικόνιση δρόμων ή ποταμών, πολύγωνα για την απεικόνιση κτηρίων ή λιμνών). Η προσέγγιση αναπαράστασης μέσω σημείων ή σχημάτων, ονομάζεται διανυσματική και οφείλει την ονομασία της στον τρόπο με τον οποίο αποθηκεύουμε τα εν λόγω δεδομένα σε επίπεδο προγραμματισμού.



(α) πραγματική αναπαράσταση (β) διανυσματική προσέγγιση (γ) προσέγγιση καμβά

Σχήμα 1.1: Αναπαράσταση χωρικών αντικειμένων [1]

Στις περισσότερες εφαρμογές, γίνεται χρήση της διανυσματικής προσέγγισης. Ωστόσο, υπάρχουν περιπτώσεις που η αναπαράσταση χαρακτηρίζεται από μεγάλη πολυπλοκότητα. Τότε προτιμάται η προσέγγιση καμβά, στην οποία κάθε αντικείμενο απεικονίζεται από ένα σύνολο εικονοστοιχείων (pixels). Χαρακτηριστικό παράδειγμα εφαρμογής αυτής της προσέγγισης, αποτελούν οι μετεωρολογικοί χάρτες, και συγκεκριμένα η καταγραφή των θερμοκρασιών. Εξαιτίας της διαφοροποίησης θερμοκρασιών που συναντάται σε μία περιοχή, είναι σχεδόν αδύνατη η ακριβής αναπαράσταση, επομένως επιλέγεται η χρήση ζωνών θερμοκρασίας, των οποίων η απεικόνιση γίνεται με διαφορετικό χρώμα πάνω στο χάρτη. Στο Σχήμα 1.1 μπορούμε να δούμε πως αναπαρίστανται τα χωρικά αντικείμενα εφαρμόζοντας και τις δύο προσεγγίσεις.

1.1.2 Σχέση μεταξύ χωρικών αντικειμένων

Όπως στους απλούς τύπους δεδομένων, έτσι και στα χωρικά αντικείμενα, η διαδικασία συλλογής και οργάνωσής τους, πραγματοποιείται με τη χρήση σχεσιακών ΒΔ. Στο Σχήμα 1.2 αναπαρίσταται ένα απλό παράδειγμα καταγραφής, όπου κάθε γραμμή αντιστοιχεί σε ένα συγκεκριμένο αντικείμενο και κάθε στήλη σε ένα γνώρισμά του, χωρικό ή μη.

City	Latitude	Longtitude	Country	ISO 2	Population
Berlin	52.5167	13.3833	Germany	DE	3,570,750
Paris	48.8566	2.3522	France	FR	11,142,303
Athens	37.9842	23.7281	Greece	GR	3,153,781
Rome	41.8931	12.4828	Italy	IT	4,297,877
Madrid	40.4167	-3.7167	Spain	ES	6,713,557
Lisbon	38.7452	-9.1604	Portugal	PT	2,986,162

Σχήμα 1.2: Καταγραφή χωρικών αντικειμένων

Το επόμενο στάδιο, μετά τη καταγραφή, είναι η αναζήτηση σχέσεων που να συνδέουν τα αντικείμενα μεταξύ τους. Για σχέσεις, στις οποίες εμπλέκονται τα χωρικά γνωρίσματα, διακρίνουμε τρεις βασικές κατηγορίες: (i) τις τοπολογικές σχέσεις, (ii) τις σχέσεις κατεύθυνσης και (iii) τις σχέσεις απόστασης . Καθεμία από αυτές, αν και προσεγγίζει με διαφορετικό τρόπο τη συσχέτιση των αντικειμένων, μπορεί να συνδυαστεί με κάποια άλλη, αν αυτό χρήζει αναγκαίο.

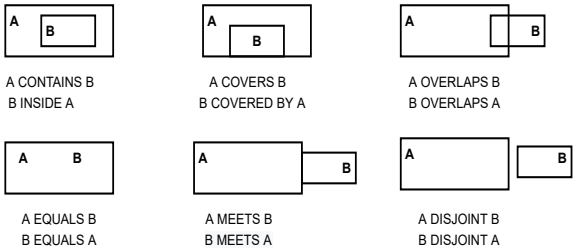
Τοπολογικές σχέσεις

Πρόκειται για ένα πλήθος σχέσεων που συνδέουν δύο αντικείμενα βάσει της λογικής των συνόλων. Η έκφραση « το Δημαρχείο είναι εντός του πάρκου » είναι μία περίπτωση τοπολογικής σχέσης, καθώς υποδηλώνει ότι η έκταση που καταλαμβάνει το Δημαρχείο είναι ένα υποσύνολο της έκτασης του πάρκου. Οι τοπολογικές σχέσεις εφαρμόζονται τόσο σε χωρικά αντικείμενα ίδιου τύπου (π.χ. μεταξύ σημείων), όσο και σε αντικείμενα διαφορετικού τύπου (π.χ. μεταξύ σημείου και πολυγώνου) [5]. Στον Πίνακα 1.1 συνοψίζονται οι συνδυασμοί συσχέτισης μεταξύ των αντικειμένων ίδιου/διαφορετικού τύπου, καθώς και οι σχέσεις που δύναται να εφαρμοστούν στην κάθε περίπτωση.

Topological Relationships	Combination of object types					
	P-P	P-L	P-Pg	L-L	L-Pg	Pg-Pg
disjoint	Y	Y	Y	Y	Y	Y
meets	-	-	-	Y	Y	Y
equals	Y	-	-	Y	-	Y
contains	-	-	Y	-	Y	Y
covers	-	Y	Y	Y	Y	Y
overlaps	-	-	-	Y	Y	Y
Y/- means the topological relationships exists/undefined						

Πίνακας 1.1: Τοπολογικές σχέσεις μεταξύ αντικειμένων [5]

Γενικά, όλες οι τοπολογικές σχέσεις εκτός της disjoint, απορρέουν από τη σχέση intersect και αποτελούν απλά εξειδικευμένες περιπτώσεις της. Για να γίνει πιο κατανοητό αυτό, ακολουθούν τα Σχήματα 1.3 και 1.4, τα οποία παρουσιάζουν όλες τις πιθανές σχέσεις μεταξύ πολυγώνων και τις μαθηματικές εκφράσεις τους, αντίστοιχα. Για την περίπτωση των πολυγώνων, αξίζει να τονιστεί, πως στη συσχέτισή τους εμπλέκονται τόσο το εσωτερικό (interior) τους, όσο και η οριοθέτησή (boundary) τους.



Σχήμα 1.3: Αναπαράσταση σχέσεων [2]

Topological Relationship	Equivalent Boundary (BDRY) / Interior (INTR) relationships
disjoint(A,B)	$(INTR(A) \cap INTR(B) = \emptyset) \wedge (BDRY(A) \cap BDRY(B) = \emptyset)$
meets(A,B)	$(INTR(A) \cap INTR(B) = \emptyset) \wedge (BDRY(A) \cap BDRY(B) \neq \emptyset)$
equals(A,B)	$(INTR(A) = INTR(B)) \wedge (BDRY(A) = BDRY(B))$
contains(A,B)	$INTR(B) \subset INTR(A)$
covers(A,B)	$INTR(B) \subset INTR(A) \wedge (\exists p \in A : p \subseteq BDRY(B))$
overlaps(A,B)	$(INTR(A) \cap INTR(B) = \emptyset) \wedge (\exists p \in A : p \nsubseteq INTR(B) \wedge p \nsubseteq BDRY(B))$ $\wedge (\exists p \in B : p \nsubseteq INTR(A) \wedge p \nsubseteq BDRY(A))$

Σχήμα 1.4: μαθηματικές εκφράσεις [3]

Σχέσεις κατεύθυνσης

Στην κατηγορία αυτή, τα αντικείμενα συνδέονται μεταξύ τους ποιοτικά και όχι μέσω μαθηματικών εκφράσεων [7]. Στην ουσία, η συσχέτιση που αναπτύσσεται, αφορά τον προσανατολισμό ενός αντικειμένου σε σχέση με ένα άλλο, βάσει κάποιων συστημάτων αναφοράς. Επομένως, όταν λέμε ότι ένα αντικείμενο βρίσκεται νότια, ανατολικά, πίσω ή αριστερά από ένα άλλο, προσδιορίζουμε μία σχέση κατεύθυνσης, ανάμεσά τους.

Σχέσεις απόστασης

Στις περιπτώσεις αυτές, τα αντικείμενα συσχετίζονται με βάση τη μεταξύ τους απόσταση, η οποία υπολογίζεται μέσω κάποιας μετρικής (π.χ. Ευκλείδεια απόσταση). Στην πραγματικότητα, η απόσταση χρησιμοποιείται για να αποτυπωθεί κάποια πληροφορία εγγύτητας μεταξύ των αντικειμένων (π.χ. αν βρίσκονται κοντά ή μακριά).

1.1.3 Χωρικά ερωτήματα

Εξ ορισμού, τα χωρικά ερωτήματα αποτελούν μεθόδους που εφαρμόζονται σε ένα ή περισσότερα σύνολα χωρικών αντικειμένων, αναζητώντας εκείνο το υποσύνολο που ικανοποιεί, ανάλογα και με τις ανάγκες του εκάστοτε αιτήματος, ορισμένες χωρικές σχέσεις. Από τη μία, ερωτήματα όπως το spatial range query ή το nearest neighbor query, αναζητούν τα αντικείμενα ενός συνόλου, που ανήκουν σε μία οριοθετημένη περιοχή ή βρίσκονται κοντά σε ένα σημείο αναφοράς αντίστοιχα. Από την άλλη, ερωτήματα όπως το spatial join, συνδυάζουν δύο σύνολα αντικειμένων, ανακτώντας το υποσύνολο του καρτεσιανού γινομένου τους, που πληροί τις ζητούμενες προϋποθέσεις.

Είναι γεγονός ότι τα σύγχρονα ΣΔΒΔ μπορούν να υποστηρίξουν τη μοντελοποίηση και την έκφραση χωρικών ερωτημάτων. Πρέπει ωστόσο να γίνει κατανοητό, πως ο στόχος δεν είναι μόνο η διεκπεραίωση αυτών των ερωτημάτων, αλλά και η αποτελεσματική αξιολόγησή τους. Στο σημείο αυτό θα αναλύσουμε τις δυσκολίες που εμφανίζονται κατά την ανάκτηση και επεξεργασία των αντικειμένων, οι οποίες κατά συνέπεια επηρεάζουν και την απόδοση των ερωτημάτων.

Πρόσβαση και ανάκτηση αντικειμένων

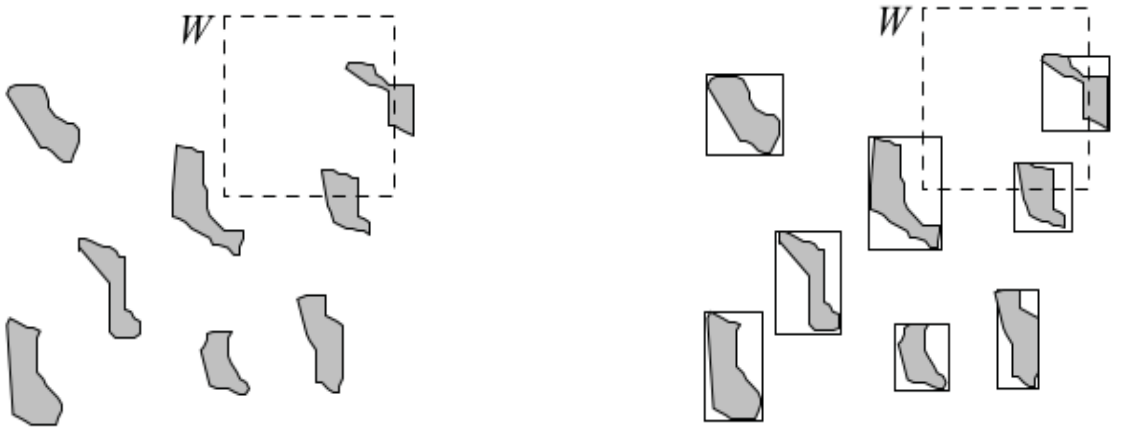
Το πρώτο ζήτημα που καλούμαστε να αντιμετωπίσουμε κατά τη διαχείριση χωρικών αντικειμένων, είναι η γρήγορη ανάκτησή τους. Λόγω της πολυδιάστατης μορφής που έχουν,

είναι αδύνατη η αποθήκευσή τους στη μνήμη, με τέτοιο τρόπο, που να εξασφαλίζεται η χωρική εγγύτητα. Κατά συνέπεια, όταν εκτελείται ένα χωρικό ερώτημα, δεν είναι εφικτό να ανακτηθεί άμεσα το σύνολο εκείνων των αντικειμένων, που εμπεριέχει τα πιθανά αποτελέσματά του.

Τη λύση στην αντιμετώπιση αυτού του ζητήματος έρχεται να δώσει η διαδικασία της χωρικής ευρετηρίασης. Αυτή η διαδικασία επιτυγχάνεται από μεθόδους που χρησιμοποιούν δομές δεδομένων, ικανές να υποστηρίξουν αλγορίθμους αναζήτησης της Υπολογιστικής Γεωμετρίας [8]. Στο κεφάλαιο 2 θα παρουσιάσουμε αναλυτικά ορισμένες τέτοιες μεθόδους, οι οποίες είναι ευρέως διαδεδομένες, καθώς και μερικές ερευνητικές λύσεις που ανά καιρούς, έχουν προταθεί.

Υπολογιστικό κόστος επεξεργασίας των αντικειμένων

Σε πολλές περιπτώσεις, τα χωρικά ερωτήματα καλούνται να εφαρμοστούν πάνω στις γεωμετρικές εκτάσεις των αντικειμένων. Εξαιτίας της πολυπλοκότητας αυτών των γεωμετριών, η διαδικασία αυτή επιφέρει κάποιο υπολογιστικό κόστος, το οποίο γίνεται ιδιαίτερα επιβαρυντικό όταν χρειάζεται να ελεγχθούν αντικείμενα που δεν είναι αποτελέσματα ενός ερωτήματος. Σε αυτές τις περιπτώσεις υποχρεούμαστε να επεξεργαστούμε αρκετές από τις ακμές τους, για να επιβεβαιώσουμε ότι δεν ανήκουν στο σύνολο των αποτελεσμάτων.



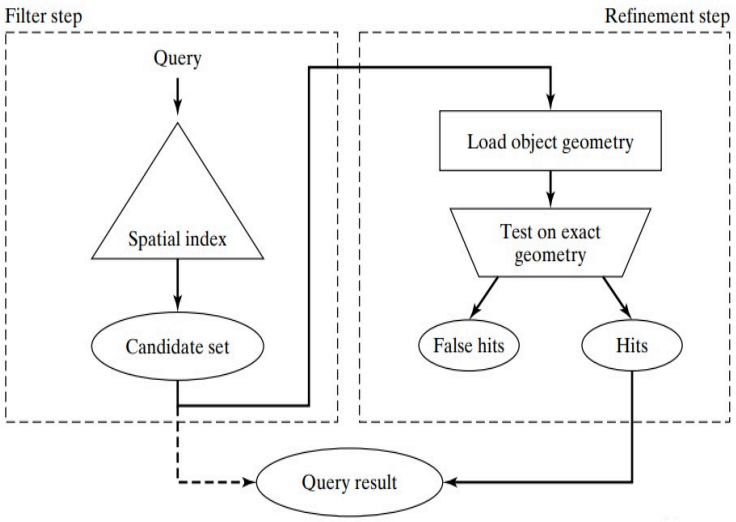
(α) Πραγματικές εκτάσεις (β) MBRs

Σχήμα 1.5: Χωρικό Ερώτημα Διαστήματος [3]

Τη λύση στη μείωση αυτής της υπολογιστικής επιβάρυνσης, έρχεται να δώσει η προσέγγιση του minimum bounding rectangle (MBR). Ως MBR εννοείται το τετράγωνο που περικλείει στον ελάχιστο δυνατό βαθμό όλη τη γεωμετρική έκταση ενός αντικειμένου. Για

κάθε MBR, η πληροφορία που συλλέγουμε, είναι οι συντεταγμένες της κάτω αριστερής και της πάνω δεξιάς γωνίας, οι οποίες είναι αρκετές για να προσδιοριστεί η θέση του στο χώρο. Στο Σχήμα 1.5, βλέπουμε πως αναπαρίσταται ένα χωρικό ερώτημα διαστήματος,τόσο με τις πραγματικές εκτάσεις των αντικειμένων, όσο και με τα MBRs τους.

Πλέον, με τη χρήση των MBRs, η εκτέλεση ενός ερωτήματος διασπάται σε δύο στάδια. Πιο αναλυτικά, σε πρώτο στάδιο εφαρμόζουμε το ερώτημα στα MBRs των αντικειμένων, για να βρούμε ένα σύνολο αποτελεσμάτων του. Έπειτα, για κάθε αντικείμενο αυτού του συνόλου, χρησιμοποιούμε την ακριβή γεωμετρία του, για να επαληθεύσουμε αν ήταν ορθή ή όχι, η επιλογή του ως πιθανό αποτέλεσμα. Στη βιβλιογραφία το πρώτο στάδιο ονομάζεται filter step, ενώ το στάδιο της επαλήθευσης refinement step. Στο Σχήμα 1.6, μπορούμε να δούμε αναλυτικά όλα τα στάδια εφαρμογής ενός ερωτήματος.



Σχήμα 1.6: Στάδια ερωτήματος [4]

Γενικά, η προαναφερθείσα διαδικασία, χαρακτηρίζεται ως ιδιαίτερα αποτελεσματική για τους εξής λόγους:

- Αν το MBR ενός αντικειμένου δεν χαρακτηρίζεται ως αποτέλεσμα του ερωτήματος, τότε δεν είναι απαραίτητος ο έλεγχος της ακριβής γεωμετρίας του, γεγονός που περιορίζει σημαντικά τους αναγκαίους υπολογισμούς.

- Το κόστος ελέγχου ενός αντικειμένου τείνει να είναι σταθερό, ανεξάρτητα με το αν ανήκει στο σύνολο των αποτελεσμάτων ή οχι.

Όπως θα δούμε και στο επόμενο κεφάλαιο, όλες οι μέθοδοι ευρετηρίασης αντικειμένων

με γεωμετρικές εκτάσεις, ασχολούνται αποκλειστικά με τα MBRs τους, εξαιτίας της φθηνής υπολογιστικής επεξεργασίας που χρειάζονται.

1.2 Αντικείμενο της διπλωματικής

Το αντικείμενο της παρούσας διπλωματικής εργασίας, αφορά την ευρετηρίαση χωρικών αντικειμένων στο επίπεδο της Κύριας Μνήμης. Προτείνουμε λοιπόν, μία μέθοδο συλλογής και οργάνωσης αποκλειστικά για MBRs αντικειμένων με εκτάσεις στο χώρο, στην οποία βασιζόμαστε σε μετέπειτα στάδιο, για να αναπτύξουμε έναν αλγόριθμο διεκπεραίωσης ερωτημάτων k πλησιέστερων γειτόνων.

1.2.1 Συνεισφορά

Η βασική συνεισφορά της διπλωματικής, συνοψίζεται ως εξής:

1. Σχεδιάζουμε μία διεπίπεδη διαμέριση του χώρου, βασισμένη στη λογική της εφαρμογής ενός πλέγματος κελιών, εισάγοντας ένα επιπλέον επίπεδο, που διαχωρίζει το περιεχόμενο του κάθε κελιού σε δεκαέξι κλάσεις.

2. Προτείνουμε μία μέθοδο ευρετηρίασης στη Κύρια Μνήμη, βάσισμένη στη διεπίπεδη διαμέριση του χώρου, δεσμεύοντας όσο το δυνατό λιγότερη μνήμη στο σύστημα, και περιορίζοντας παράλληλα το πλήθος ελέγχων που χρειάζονται για την ορθή αποθήκευση των αντικειμένων.

3. Αναπτύσσουμε έναν αλγόριθμο για τη διεκπεραίωση ερωτημάτων k πλησιέστερων γειτόνων, ο οποίος συνδέεται άμεσα με τον τρόπο που διαχειριζόμαστε το χώρο και την καταγραφή των δεδομένων του.

4. Εισάγουμε μία μέθοδο που εξαλείφει το ενδεχόμενο εμφάνισης διπλότυπων αντικειμένων, στα αποτελέσματα ερωτημάτων.

5. Μελετάμε τέσσερις μεθόδους σχετικά με την αποφυγή επαναλαμβανόμενης επίσκεψης στα ίδια κελιά του χώρου. Από την εφαρμογής τους στον αλγόριθμό μας, καταλήγουμε στο συμπέρασμα ότι η τέταρτη μέθοδος είναι αυτή που προσφέρει την καλύτερη δυνατή επίδοση.

6. Αξιολογούμε τόσο το στάδιο της ευρετηρίασης, όσο και το στάδιο των ερωτημάτων, συγκρίνοντας τους μέσους χρόνους μας, έναντι των αντίστοιχων που προκύπτουν για δύο παραλλαγές του R-tree. Τα αποτελέσματα των δοκιμών, πάνω σε πραγματικά σύνολα δεδομένων, αποδεικνύουν την υπεροχή μας και στα δύο στάδια της πειραματικής μελέτης.

1.3 Οργάνωση του τόμου

Τα υπόλοιπα κεφάλαια αυτής της εργασίας οργανώνονται ως εξής:

- **Κεφάλαιο 2 (Βασικά Στοιχεία Χωρικής Ευρετηρίασης)**: Βασικές γνώσεις σχετικά με τις κατηγορίες χωρικών ευρετηριάσεων.
- **Κεφάλαιο 3 (Προτεινόμενη Υλοποίηση)**: Εκτενής παρουσίαση όλων των σταδίων ανάπτυξης της υλοποίησής μας για την ευρετηρίαση αντικειμένων και τη διεκπεραίωση ερωτημάτων k πλησιέστερων γειτόνων.
- **Κεφάλαιο 4 (Πειραματική Αξιολόγηση)**: Αξιολόγηση των διαφορετικών εκδόσεων της υλοποίησής μας και σύγκριση των αποτελεσμάτων μας έναντι υπάρχουσας προσέγγισης.
- **Κεφάλαιο 5 (Συμπεράσματα)**: Συμπεράσματα σχετικά με τη παρούσα εργασία και μελλοντικές επεκτάσεις της.

Κεφάλαιο 2

Βασικά Στοιχεία Χωρικής Ευρετηρίασης

Στο κεφάλαιο αυτό ασχολούμαστε εκτενώς με τις τεχνικές χωρικής ευρετηρίασης. Πιο αναλυτικά, περιγράφουμε τη λογική πίσω από την ανάπτυξη των μεθόδων χωρικής πρόσβασης, παρουσιάζοντας τις κατηγορίες στις οποίες διακρίνονται και εξηγώντας τυχόν πλεονεκτήματα/μειονεκτήματα της καθεμίας. Επιπλέον, κάνουμε μια μικρή αναφορά σε διαδεδομένες μεθόδους που κατατάσσονται στην κάθε κατηγορία.

2.1 Μέθοδοι Χωρικής Πρόσβασης

Όπως είδαμε και στο προηγούμενο κεφάλαιο, η μεγαλύτερη πρόκληση που καλείται κανείς να διαχειριστεί κατά την ενασχόληση με ερωτήματα χωρικού τύπου, είναι η αποδοτική πρόσβαση στα δεδομένα. Εδώ και δεκαετίες, η ερευνητική κοινότητα έστρεψε την προσοχή της στην αναζήτηση λύσεων που αντιμετωπίζουν το ζήτημα της αποτελεσματικής καταγραφής αντικειμένων, με πολυδιάστατη μορφή στο χώρο. Αυτή η απόπειρα, είχε ως αποτέλεσμα την εμφάνιση των πρώτων μεθόδων χωρικής πρόσβασης [9, 10].

Ο πρωταρχικός στόχος αυτών των μεθόδων αφορούσε την αποθήκευση των δεδομένων στο επίπεδο του δίσκου. Παρόλα αυτά, με την εξέλιξη των υπολογιστικών συστημάτων, ενισχύθηκε η επιθυμία της καταγραφής των αντικειμένων με όσο το δυνατό λιγότερες I/O αλληλεπιδράσεις, αξιοποιώντας στην ουσία το επίπεδο της Κύριας Μνήμης [11, 12]. Ανεξάρτητα από το επίπεδο μνήμης που επιλέγεται, κάθε μέθοδος χωρικής πρόσβασης αναπτύσσεται, ακουλουθώντας μία διαδικασία δύο σταδίων. Στο πρώτο στάδιο, πραγματοποιείται μία διαμέριση του χώρου, με σκοπό να δημιουργηθούν ομάδες αντικειμένων, που εντοπίζονται σε κοντινές θέσεις και αποθηκεύονται στο ίδιο μπλοκ του ευρετηρίου (ας υποθέσουμε ως μπλοκ,

με σελίδα του δίσκου). Στο δεύτερο στάδιο, πραγματοποιείται η οργάνωση (μονοεπίπεδη ή ιεραρχική) αυτών των μπλοκ στο εν λόγω ευρετήριο.

Γενικά, όλες οι μέθοδοι ευρετηρίασης χωρίζονται σε δύο βασικές κατηγορίες: τις Space-oriented Partitioning (SOP) και τις Data-oriented Partitioning (DOP) ευρετηριάσεις. Η διαφοροποίηση μεταξύ των δύο κατηγοριών συνδέεται με τον τρόπο που επιλέγεται να γίνεται η διαμέριση του χώρου [13]. Στο υπόλοιπο κεφάλαιο λοιπόν, θα ασχοληθούμε αποκλειστικά με αυτές τις κατηγορίες, παρουσιάζοντας τις πιο γνωστές μεθόδους τους.

2.2 SOP μέθοδοι ευρετηρίασης

Αυτή η κατηγορία εμφανίσθηκε στην προσπάθεια της αποτελεσματικής συλλογής δεδομένων, που αντιπροσωπεύουν τις θέσεις των αντικειμένων (δηλ. σημεία) στο χώρο. Σε όλες τις μεθόδους επικρατεί η λογική της διαμέρισης του χώρου με τέτοιο τρόπο, ώστε να προκύπτουν τμήματα, τα οποία είναι ανεξάρτητα μεταξύ τους και φέρουν το δικό τους πλήθος αντικειμένων. Από τη μία, η πιο απλή μέθοδος, που εντάσσεται στις SOP ευρετηριάσεις, είναι η χρήση ενός πλέγματος, όπου ο χώρος διαμερίζεται ομοιόμορφα, σε κελιά αντικειμένων. Από την άλλη, στη λογική της ιεραρχικής οργάνωσης, οι πιο γνωστές μέθοδοι που ανήκουν σε αυτή την κατηγορία, είναι η k-d tree [14] και η quad-tree [15].

Γενικά, οι συγκεκριμένες μέθοδοι προτιμώνται ιδιαίτερα όταν το ζητούμενο είναι η ευρετηρίαση των αντικειμένων στο επίπεδο της Κύριας Μνήμης, λόγω της αποτελεσματικότητά τους, στις αναζητήσεις και στις ενημερώσεις εντός του ευρετηρίου. Επιπλέον, η ανεξαρτησία μεταξύ των διαμερισμένων τμημάτων του χώρου, τις καθιστά ικανές για την εφαρμογή τους σε παράλληλα και κατανεμημένα συστήματα [16, 17, 18].

Πέρα, όμως από τα προαναφερθέντα πλεονεκτήματά τους, παρουσιάζουν ένα βασικό μειονέκτημα, όταν εφαρμόζονται στην αποθήκευση αντικειμένων με γεωμετρικές εκτάσεις στο χώρο. Πιο αναλυτικά, στις περιπτώσεις αυτές, καθώς τα αντικείμενα καταλήγουν να επικαλύπτουν πολλαπλά διαμερισμένα τμήματα, είναι αναγκαία η ξεχωριστή ανάθεσή τους σε καθένα από αυτά, μέσω της δημιουργίας αντίγραφων. Από τη μία, η διαδικασία αυτή εγγυάται την άμεση αποθήκευση των αντικειμένων, με όσο το δυνατό λιγότερους ελέγχους στο στάδιο της ευρετηρίασης. Από την άλλη όμως, είναι πιθανό να επηρεάζεται η ορθότητα των ερωτημάτων, καθώς ένα ή περισσότερα αντικείμενα μπορεί να χαρακτηριστεί αποτέλεσμά του, περισσότερες από μία φορές.

Για την αποφυγή διπλότυπων εμφανίσεων των αντικειμένων στα αποτελέσματα των ερωτημάτων, έχουν προταθεί διάφορες τεχνικές. Στην πλειονότητάς τους, αφορούν ερωτήματα χωρικού διαστήματος και χωρικής σύνδεσης, και εφαρμόζονται μετά τον υπολογισμό των αποτελεσμάτων, αναζητώντας τυχόν επαναλαμβανόμενες εμφανίσεις εντός αυτού του συνόλου [19, 20]. Παρόλα αυτά, έχει προταθεί μία ακόμη τεχνική [21], στην οποία η διαχείριση των διπλότυπων γίνεται κατά το στάδιο της ευρετηρίασης, με την ανάθεση χαρακτηρισμών στα αντίγραφα ενός αντικειμένου.

2.3 DOP μέθοδοι ευρετηρίασης

Στην κατηγορία αυτή, η χωρική διαμέριση που πραγματοποιείται, επιτρέπει στα προκύπτοντα τμήματα να μπορούν να επικαλύπτονται μεταξύ τους, διασφαλίζοντας με αυτό τον τρόπο, ότι το καθένα έχει το δικό του ανεξάρτητο περιεχόμενο (δηλ. κάθε αντικείμενο ανήκει σε ένα μόνο τμήμα). Βάσει αυτού, οι συγκεκριμένες μέθοδοι συνήθως προτιμώνται στις περιπτώσεις συλλογής αντικειμένων με γεωμετρικές εκτάσεις, προκειμένου να αποφεύγεται η δημιουργία αντίγραφων, διαδικασία που είναι απαραίτητη στις SOP μεθόδους.

Η πιο διαδεδομένη μέθοδος που εμπίπτει σε αυτή την κατηγορία ευρετηριάσεων, είναι το R-tree[22]. Πρόκειται για ένα ισοζυγισμένο δέντρο, που γενικεύει ττην περίπτωση του B+-tree [23], στον πολυδιάστατο χώρο. Σε όλες τις περιπτώσεις των επιπέδων του, η λογική είναι να αποθηκεύονται MBRs δεδομένα. Συγκεκριμένα, στους κόμβους-φύλλα του, γίνεται η καταγραφή των MBRs, που χαρακτηρίζουν τα αντικείμενα του χώρου. Στους υπόλοιπους κόμβους των άλλων επιπέδων, αποθηκεύονται πάλι MBRs, τα οποία αυτή τη φορά αντιστοιχούν στα διαμερισμένα τμήματα του χώρου.

Πάνω στη λογική του R-tree, έχουν βασιστεί πολλές μέθοδοι, οι οποίες στην ουσία αποτελούν παραλλαγές του. Από τη μία, το R*-tree [24], είναι μία τέτοια περίπτωση μεθόδου, η οποία διαχειρίζεται αποτελεσματικά καταστάσεις που παραπέμπουν σε δυναμικές ενημερώσεις των δεδομένων, πραγματοποιώντας την καταγραφή τους, με όσο το δυνατό λιγότερες επικαλύψεις. Από την άλλη, το CR-tree [25], αποτελεί μία βελτιστοποιημένη έκδοση του R-tree, για την ευρετηρίαση των αντικειμένων στο επίπεδο της Κύριας Μνήμης, μιας και η πρωταρχική προσέγγιση παραπέμπει στην καταγραφή τους στο δίσκο, με στόχο τις ελάχιστες δυνατές I/O αλληλεπιδράσεις, στη διαδικασία εκτέλεσης των ερωτημάτων.

Κεφάλαιο 3

Προτεινόμενη Υλοποίηση

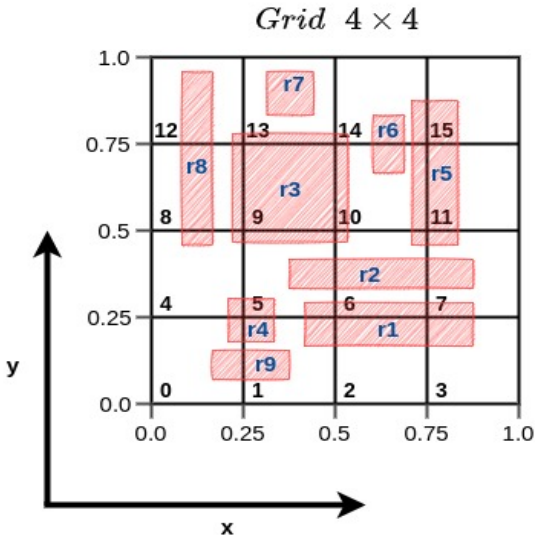
Στο κεφάλαιο αυτό, περιγράφουμε αναλυτικά, όλα τα βήματα της υλοποίησής μας. Αρχικά, στην ενότητα 3.1 αναφέρουμε την τεχνική που ακολουθούμε για τη διαμέριση του χώρου. Στη συνέχεια, με την ενότητα 3.2 παρουσιάζουμε εκτενώς τα στάδια της ευρετηρίασης των αντικειμένων στην Κύρια Μνήμη. Τέλος, στην ενότητα 3.3 εξηγούμε την αλγοριθμική διαδικασία που ακολουθούμε, για τη διεκπεραίωση ερωτημάτων k πλησιέστερων γειτόνων.

3.1 Διαμέριση του χώρου-Χαρακρηρισμός αντικειμένων

Στο πλαίσιο αυτής της υλοποίησης, η μέθοδος χωρικής πρόσβασης που επιλέγουμε να αναπτύξουμε, βασίζεται στη λογική των SOP χωρικών ευρετηριάσεων, και συγκεκριμένα στη διαχείριση του χώρου ως ένα τετράγωνο και ομοιόμορφα διαμερισμένο πλέγμα, όπου κάθε διαμέρισή (ή κελί) του, εμπεριέχει το σύνολο των αντικειμένων που του αντιστοιχούν. Από τη μία, η επιλογή αυτής της προσέγγισης, λειτουργεί υποστηρικτικά στην προσπάθειά μας, για ευρετηρίαση στο επίπεδο Κύριας Μνήμης. Από την άλλη όμως, η ενασχόληση μας με MBRs αντικειμένων, μάς υποχρεώνει να αντιμετωπίσουμε το βασικότερο μειονέκτημά της, που είναι η εμφάνιση διπλότυπων (βλπ ενότητα 2.2).

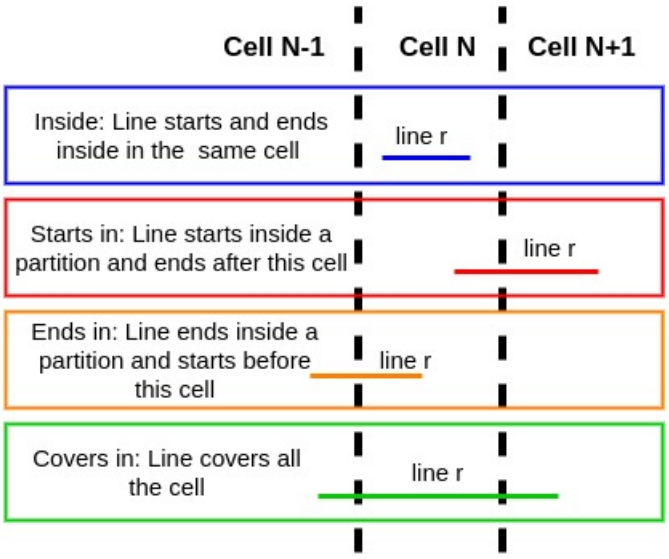
Για να γίνει πιο κατανοητό το πρόβλημα, ας θεωρήσουμε τα εννιά αντικείμενα που απεικονίζονται στο Σχήμα 3.1, τα οποία είναι τοποθετημένα σε ένα πλέγμα 4×4. Τα περισσότερα, λόγω της έκτασής τους, καταλήγουν να αντιστοιχίζονται σε πολλαπλά κελιά. Κατά συνέπεια, στην εφαρμογή ενός ερωτήματος, τα επαναλαμβανόμενα αντικείμενα, είναι πιθανό να χαρακτηριστούν ως αποτελέσματά του, περισσότερες από μία φορές. Για παράδειγμα, αν το αντικείμενο r2 ανήκει κανονικά στα αποτελέσματα ενός ερωτήματος, και εξεταστούν και τα

τρία κελιά στα οποία αντιστοιχεί (δηλ. τα κελιά 5, 6 και 7), ισάριθμες θα είναι τελικά και οι φορές που θα χαρακτηριστεί σαν αποτέλεσμα.



Σχήμα 3.1: Αναπαράσταση πλεγματικής διαμέρισης

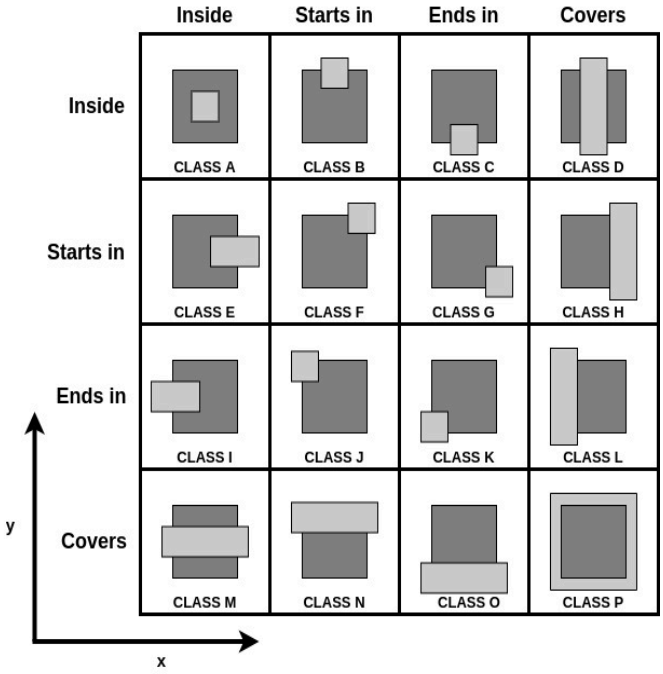
Για την αποφυγή διπλότυπων, οι περισσότερες τεχνικές που έχουν προταθεί, εφαρμόζονται στο στάδιο εκτέλεσης των ερωτημάτων, όπου αναζητούνται τυχόν επαναλαμβανόμενες εμφανίσεις και εξαλείφονται από το σύνολο των αποτελεσμάτων. Η δική μας πρόταση προσανατολίζεται στη διαχείριση των διπλότυπων, σε προγενέστερο στάδιο και συγκεκριμένα, κατά τη διαδικασία της ευρετηρίασης. Για το σκοπό αυτό εισάγουμε, μαζί με την πλεγματική διαμέριση, ένα επιπλέον επίπεδο, στο οποίο πραγματοποιούμε ένα χαρακτηρισμό-κατηγοριοποίηση των αντικειμένων.



Σχήμα 3.2: Περιπτώσεις έκτασης των αντικειμένων στον άξονα x

Μέσω του χαρακτηρισμού, αυτό που επιδιώκεται είναι ο διαχωρισμός των αντικειμένων, ανάλογα με το πώς εκτείνονται σε σχέση με το κελί, στο οποίο ανήκουν. Στο σχήμα 3.2 βλέπουμε τις τέσσερις πιθανές εκτάσεις των αντικειμένων ενός κελιού N, στον άξονα x. Ειδικότερα, ένα αντικείμενο μπορεί:

- Να αρχίζει και να τελειώνει εσωτερικά του κελιού N (μπλε περίγραμμα)
- Να αρχίζει εντός του κελιού N και να εκτείνεται προς τα δεξιά του (κόκκινο περίγραμμα)
- Να ξεκινάει από τα αριστερά του κελιού N και να τελειώνει εντός του (κίτρινο περίγραμμα)
- Να εκτείνεται με τέτοιο τρόπο ώστε να υπερκαλύπτει όλη την έκταση του κελιού N (πράσινο περίγραμμα)



Σχήμα 3.3: Χαρακτηρισμοί των αντικειμένων

Με την ίδια λογική, οι ίδιες περιπτώσεις εμφανίζονται και στις εκτάσεις των αντικειμένων, στον άξονα y. Επομένως, συνδυάζοντας όλες τις δυνατές περιπτώσεις και στις δύο διαστάσεις, καταλήγουμε σε δεκαέξι διαφορετικούς χαρακτηρισμούς, οι οποίοι συνοψίζονται στο Σχήμα 3.3. Επικαλούμενοι ξανά την αναπαράσταση του Σχήματος 3.1, μπορούμε

να δούμε συγκεντρωτικά το σύνολο των αντικειμένων κάθε κελιού, με και χωρίς τους χαρακτηρισμούς, στον πίνακα που ακολουθεί (Πίνακας 3.1). Σε όλο το υπόλοιπο κείμενο, θα αναφερόμαστε σε αυτούς τους χαρακτηρισμούς με τον όρο « κλάσεις ».

Cell	Primary Partitioning	Secondary Partitioning
0	{ r4, r9 }	E = { r9 }, F = { r4 }
1	{ r1, r4, r9 }	F = { r1 }, I = { r9 }, J = { r4 }
2	{ r1 }	N = { r1 }
3	{ r1 }	J = { r1 }
4	{ r3, r4, r8 }	B = { r8 }, F = { r3 }, G = { r4 }
5	{ r1, r2, r3, r4 }	E = { r2 }, G = { r1 }, K = { r4}, N = { r3 }
6	{ r1, r2, r3, r5 }	F = { r5 }, J = { r3 }, M = { r2 }, O = { r1 }
7	{ r1, r2, r5 }	I = { r2 }, J = { r5}, K = { r1 }
8	{ r3, r8 }	D = { r8 }, H = { r3 }
9	{ r3 }	P = { r3 }
10	{ r3, r5, r6 }	B = { r6 }, H = { r5 }, L = { r3 }
11	{ r5 }	L = { r5 }
12	{ r3, r8 }	C = { r8 }, G = { r3 }
13	{ r3, r7 }	A = { r7 }, O = { r3 }
14	{ r3, r5, r6 }	C = { r6 }, G = { r5 }, K = { r3 }
15	{ r5 }	K = { r5 }

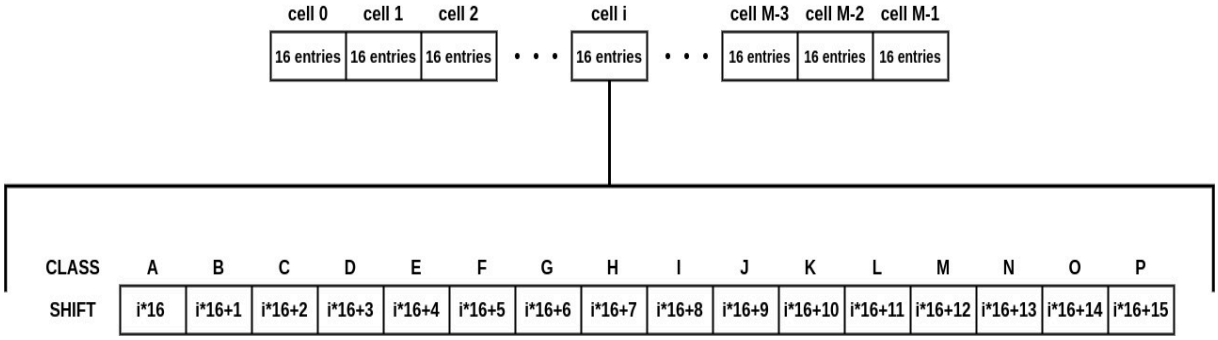
Πίνακας 3.1: Σύνολο αντικειμένων ανά κελί, με/χωρίς χαρακτηρισμούς

3.2 Χωρική ευρετηρίαση στη Κύρια Μνήμη

Το επόμενο στάδιο, μετά τη διεπίπεδη διαμέριση του χώρου μας, είναι η ευρετηρίαση των αντικειμένων. Η διαδικασία αυτή περιγράφεται συνοπτικά από τα εξής βήματα:

- Βήμα 1^o : Αναζήτηση των κελιών, που αντιστοιχεί το κάθε αντικείμενο, και ανάθεση του κατάλληλου χαρακτηρισμού (κλάση)
- Βήμα 2^o : Υπολογισμός του συνόλου των αντικειμένων, για κάθε κελί του πλέγματος, και δέσμευση του χώρου αποθήκευσής τους
- Βήμα 3^o : Τοποθέτηση των αντικειμένων στα συσχετιζόμενα κελιά

Όπως ήδη έχει αναφερθεί, μάς ενδιαφέρει η ευρετηρίαση να γίνεται στο επίπεδο της Κύριας Μνήμης. Για την επίτευξη αυτού του σκοπού, χρησιμοποιούμε δύο πίνακες δυναμικού μεγέθους. Ο πρώτος από αυτούς, είναι ένας πίνακας ακεραίων, μεγέθους $16 \times M^2$ (όπου M^2 το πλήθος των κελιών), ο οποίος έχει βοηθητικό ρόλο. Στην ουσία, κάθε δεκαέξι συνεχόμενες θέσεις του αντιστοιχίζονται με ένα από τα κελιά του πλέγματος, και λειτουργούν ως μετρητές που καταγράφουν το πλήθος των αντικειμένων που χαρακτηρίζονται από την εκάστοτε κλάση. Αυτό ακριβώς απεικονίζεται στο Σχήμα 3.4, που δείχνει τη μορφή των συνεχόμενων θέσεων ενός κελιού i.



Σχήμα 3.4: Πίνακας μετρητών

Ο δεύτερος και πιο βασικός πίνακας, πρόκειται για το χώρο αποθήκευσης των αντικειμένων. Πιο αναλυτικά, πρόκειται για ένα πίνακα μεγέθους M^2, στον οποίο κάθε θέση (που αντιπροσωπεύει ένα κελί του πλέγματος) αποτελεί ένα διάνυσμα, ίδιου τύπου, με αυτόν που χρησιμοποιούμε για την έκφραση των αντικειμένων σε επίπεδο προγραμματισμού. Έχοντας πλέον μία εικόνα των δομών που χρησιμοποιούμε, είμαστε σε θέση να δούμε εκτενώς τα βήματα της προτεινόμενης ευρετηρίασης.

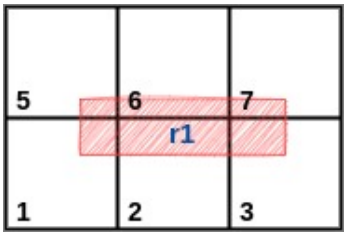
3.2.1 Αναζήτηση κελιών-Ανάθεση χαρακτηρισμών

Από τη στιγμή που ασχολούμαστε με αντικείμενα που πιθανώς εκτείνονται σε περισσότερα από ένα κελιά, η πρώτη ενέργεια που πρέπει να κάνουμε είναι να τα εντοπίσουμε. Έτσι λοιπόν, για κάθε αντικείμενο του χώρου, χρησιμοποιούμε τις συντεταγμένες του MBR του, μέσω των οποίων βρίσκουμε το πρώτο (κάτω αριστερά) και το τελευταίο (πάνω δεξιά) κελί που το περικλείουν, ενώ ταυτόχρονα αποκτούμε γνώση για την έκταση που καταλαμβάνει σε καθεμία από τις δύο διαστάσεις.

Αξιοποιώντας αυτή την πληροφορία, εφαρμόζουμε διαπέραση αυτών των κελιών κατά γραμμές. Για κάθε κελί που συναντάμε, οι πράξεις που χρειάζεται να κάνουμε είναι δύο.

Αρχικά, αναζητούμε, ανάμεσα στις δεκαέξι κλάσεις, αυτή που χαρακτηρίζει κατάλληλα το υπό διερεύνηση αντικείμενό μας. Έπειτα, βάσει της κλάσης που επιλέγεται, προχωράμε στον υπολογισμό της μετατόπισης που χρειάζεται να γίνει εντός του πίνακα μετρητών, ώστε να αυξήσουμε επιτυχώς τη τιμή της θέσης που μάς απασχολεί.

Για να γίνει κατανοητή η διαδικασία που μόλις περιγράψαμε, στο Σχήμα 3.5α παρουσιάζουμε μεμονωμένα το αντικείμενο r1 από την αναπαράσταση του Σχήματος 3.1. Το συγκεκριμένο αντικείμενο, όπως φαίνεται, καταλαμβάνει ένα χώρο κελιών μεγέθους 2×3, ξεκινώντας εντός του κελιού 1, και φτάνοντας μέχρι και το εσωτερικό του κελιού 7. Οι εντοπισμοί των κατάλληλων κλάσεων, καθώς και ο υπολογισμοί των απαραίτητων μετατοπίσεων πραγματοποιούνται κατά γραμμές στο χώρο, δηλαδή πρώτα για τα κελιά 1,2 και 3, και στη συνέχεια για τα κελιά 5, 6 και 7. Οι κλάσεις και οι μετατοπίσεις, που επιλέγονται κατά τη διαπέραση των κελιών, συγκεντρώνονται στο Σχήμα 3.5β.



(α)

Cell Id	Class	Shift
1	F	1 * 16 + 5
2	N	2 * 16 + 13
3	J	3 * 16 + 9
5	G	5 * 16 + 6
6	O	6 * 16 + 14
7	K	7 * 16 + 10

(β)

Σχήμα 3.5: Παράδειγμα εντοπισμού κλάσεων & υπολογισμού μετατοπίσεων

3.2.2 Υπολογισμός συνόλου αντικειμένων-Δέσμευση χώρου

Μετά το στάδιο της καταμέτρησης των αντικειμένων, που ανήκουν στη κάθε κλάση ενός κελιού, σειρά έχει η δέσμευση του χώρου στον πίνακα αποθήκευσης. Για το σκοπό αυτό, εκτελούμε έναν εμφωλευμένο βρόχο, στον οποίο, για κάθε κελί του πλέγματος, αθροίζουμε τις τιμές των δεκαέξι θέσεων από τον πίνακα μετρητών, που του αντιστοιχούν. Στη συνέχεια, χρησιμοποιούμε το τελικό αποτέλεσμα των αθροισμάτων, ώστε να δημιουργήσουμε στην επιθυμητή θέση του πίνακα αποθήκευσης, ένα διάνυσμα τέτοιας χωρητικότητας.

Η συγκεκριμένη διαδικασία είναι απλή ως προς το κομμάτι υλοποίησης, έχοντας σταθερό υπολογιστικό κόστος, αφού χρειάζονται δεκαέξι εντολές πρόσθεσης και μία εντολή δέσμευ-

σης, για την περίπτωση του κάθε κελιού. Πέρα όμως από την απλότητά της, συμβάλλει στην ουσιώδη αξιοποίηση της μνήμης που δεσμεύεται, δίνοντας λύση στα ζητήματα που ακολουθούν.

Κατανομή αντικειμένων στο χώρο

Όταν ασχολούμαστε με την ευρετηρίαση ενός συνόλου αντικειμένων, δεν είναι δεδομένο ότι αυτό ακολουθεί ομοιόμορφη κατανομή στο χώρο. Κατά συνέπεια, ενισχύεται η πιθανότητα, τα κελιά του πλέγματος να παρουσιάζουν αποκλίσεις ως προς το πλήθος των αντικειμένων που περιέχουν στο εσωτερικό τους. Μάλιστα, δεν αποκλείεται και το χειρότερο σενάριο, στο οποίο ορισμένα από αυτά παραμένουν τελείως κενά.

Αντίγραφα αντικειμένων

Όπως έχει ήδη αναλυθεί, για οποιοδήποτε αντικείμενο, που εκτείνεται σε περισσότερα από ένα κελιά, επιλέγουμε να δημιουργήσουμε ισάριθμα αντίγραφά του, αποθηκεύοντας τα στις κατάλληλες θέσεις-κελιά του πίνακα αποθήκευσης. Από τη στιγμή όμως που η διαμέριση αποτελεί μεταβλητή παράμετρο του προγράμματος, είμαστε σε θέση να προσδιορίσουμε τον ακριβή αριθμό αυτών των αντιγράφων, αφότου μεσολαβήσει το βήμα της καταμέτρησης (βλπ προηγούμενη υποενότητα).

Συνοψίζοντας λοιπόν, με τη προτεινόμενη διαδικασία για τη δέσμευση μνήμης, κατορθώνουμε να διαχειριστούμε τα προαναφερόμενα ζητήματα, διασφαλίζοντας ότι κάθε κελί:

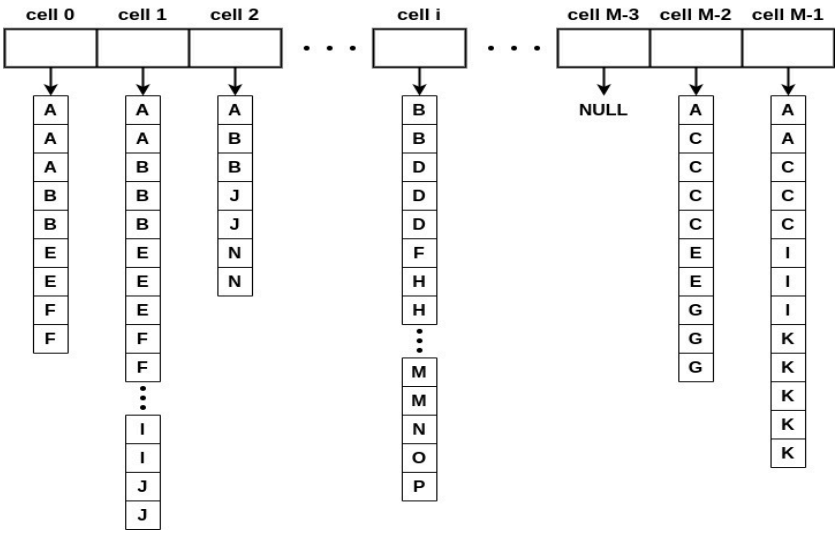
- Έχει το χώρο που επαρκεί για την αποθήκευση όλων των αντικειμένων του
- Δεν έχει επιπλέον θέσεις, ανεκμετάλλευτες καθόλη την εκτέλεση του προγράμματος
- Συμπεριλαμβάνει όλα τα αντίγραφα των αντικειμένων στα οποία εμπλέκεται, αλλά δεν ανήκουν εξ ολοκλήρου στο εσωτερικό του

3.2.3 Τοποθέτηση των αντικειμένων

Στο βήμα αυτό, ακολουθούμε την ίδια λογική με αυτή της υποενότητας 3.2.1. Για κάθε αντικείμενο που εξετάζουμε, αφού βρούμε τα κελιά στα οποία αντιστοιχίζεται, πραγματοποιούμε τη τοποθέτηση των αντιγράφων του, στα διανύσματα των κατάλληλων θέσεων του πίνακα αποθήκευσης. Όπως φαίνεται στο Σχήμα 3.6, επιλέγουμε τα αντικείμενα του κάθε

κελιού, να αποθηκεύονται ομαδοποιημένα ανάλογα την κλάση στην οποία ανήκουν, ξεκινώντας με αντικείμενα της κλάσης A, μετά της κλάσης B, της κλάσης C κ.ο.κ.

Αυτός ο τρόπος αποθήκευσης είναι εφικτός, χάρις τον πίνακα μετρητών. Στην ουσία, κάθε τιμή αυτού του πίνακα, μάς υποδεικνύει το πλήθος των θέσεων στο διάνυσμα του εκάστοτε κελιού, για τη συνεχόμενη καταγραφή των αντικειμένων μίας κλάσης. Επιπλέον, μέσω του ίδιου πίνακα, εντοπίζεται το σημείο από το οποίο αρχίζει η εισαγωγή των αντικειμένων της κάθε κλάσης, αθροίζοντας απλά τους μετρητές των κλάσεων που έχουν προηγηθεί. Έτσι, γνωρίζουμε ότι τα αντικείμενα κλάσης A αρχίζουν από τη πρώτη θέση του διανύσματος, τα αντικείμενα κλάσης B από τη θέση που ορίζει ο μετρητής της A, τα αντικείμενα της C από το άθροισμα των μετρητών των A και B κ.ο.κ.



Σχήμα 3.6: Πίνακας αποθήκευσης

3.3 Αλγοριθμική διαδικασία ερωτημάτων

Με την ολοκλήρωση της χωρικής ευρετηρίασης, το μόνο κομμάτι της υλοποίησης που απομένει να δούμε, είναι η ανάπτυξη των χωρικών ερωτημάτων. Στην περίπτωσή μας, επικεντρωνόμαστε αποκλειστικά στη διεκπεραίωση ερωτημάτων εύρεσης πλησιέστερων γειτόνων. Πιο αναλυτικά, δοθέντος ενός σημείου αναφοράς q, αυτό που πραγματοποιούμε, είναι ο εντοπισμός των αντικειμένων που χαρακτηρίζονται ως τα πιο κοντινά του.

3.3.1 Υπολογισμός αποστάσεων

Σε τέτοιου είδους ερωτήματα, η πιο διαδεδομένη μετρική που χρησιμοποιείται είναι η Ευκλείδεια απόσταση, μέσω της οποίας υπολογίζεται η πραγματική απόσταση ενός αντικειμένου, από το σημείο που εξετάζουμε. Η μαθηματική σχέση που περιγράφει αυτό τον υπολογισμό έχει την ακόλουθη μορφή:

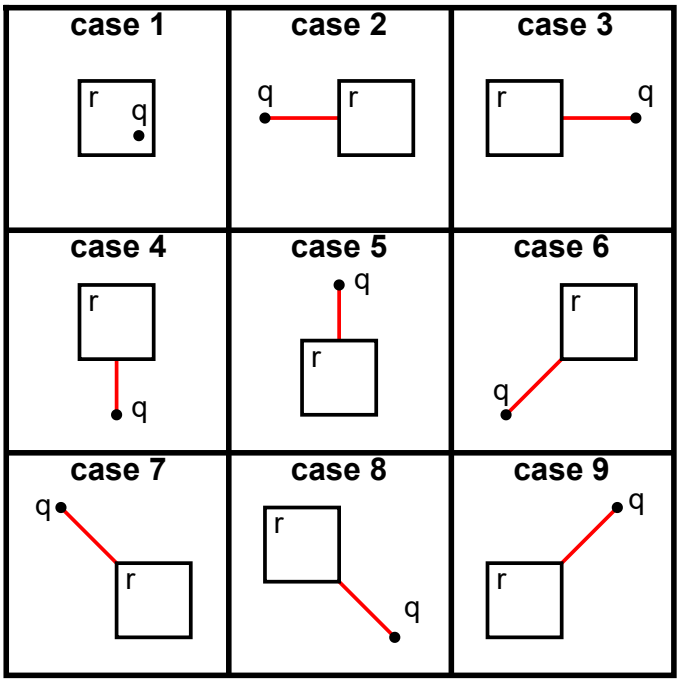
$$Distance\_metric = \sqrt{(q.x - x)^2 + (q.y - y)^2} \tag{3.1}$$

Στην υλοποίησή μας, ο υπολογισμός των αποστάσεων, λειτουργεί κυρίως ως κριτήριο σύγκρισης μεταξύ των αντικειμένων. Στην ουσία, χρησιμοποιούμε αυτές τις τιμές για να ταξινομήσουμε κατά φθίνουσα σειρά τα αντικείμενα που εξετάζουμε, κίνηση που διευκολύνει στη συνέχεια τον εντοπισμό εκείνων που χαρακτηρίζονται ως πλησιέστερα του σημείου αναφοράς. Έτσι λοιπόν, αντί να επιβαρυνόμαστε με επιπλέον υπολογιστικό κόστος από την ακρίβεια που παρέχει η Ευκλείδεια απόσταση, επιλέγουμε να εφαρμόσουμε μία παραλλαγή της, μέσω της οποίας υπολογίζουμε τα τετράγωνα των αποστάσεων. Επομένως η σχέση 3.1, μετασχηματίζεται ως εξής:

$$Distance\_metric = (q.x - x)^2 + (q.y - y)^2 \tag{3.2}$$

Γενικά, στον υπολογισμό της απόστασης, εμπλέκονται δύο ζεύγη συντεταγμένων: το πρώτο, πρόκειται για τη θέση του σημείου αναφοράς, ενώ το δεύτερο αφορά το αντικείμενο, που κάθε φορά εξετάζεται. Στην περίπτωση αντικειμένων που αναπαρίστανται ως πολύγωνα, η επιλογή του κατάλληλου ζεύγους συντεταγμένων, εξαρτάται άμεσα από τη θέση τους σε σχέση με το εν λόγω σημείο. Με βάση αυτή την εξάρτηση, προκύπτουν εννιά διαφορετικές περιπτώσεις, οι οποίες συγκεντρώνονται στο Σχήμα 3.7.

Στην πρώτη από αυτές, το σημείο αναφοράς βρίσκεται σε θέση, η οποία καταλαμβάνεται από το υπό διερεύνηση αντικείμενο, οπότε θεωρούμε ότι έχουν μηδενική απόσταση μεταξύ τους. Στις επόμενες τέσσερις περιπτώσεις, βλέπουμε πως η πιο κοντινή απόσταση ισοδυναμεί με την οριζόντια (περιπτώσεις 2 και 3) ή την κάθετη (περιπτώσεις 4 και 5) προβολή του σημείου πάνω στο αντικείμενο. Τέλος, στις περιπτώσεις 6, 7, 8 και 9, η ζητούμενη απόσταση ισούται με το τετράγωνο της υποτείνουσας του νοητού τριγώνου, που σχηματίζουν οι δοθείσες συντεταγμένες. Οι ακριβείς συνθήκες, καθώς και οι τελικές εξισώσεις και των εννιά περιπτώσεων, συνοψίζονται στο Σχήμα3.8.



Σχήμα 3.7: Αναπαράσταση περιπτώσεων απόστασης μεταξύ ορθογώνιου και σημείου

Cases	Conditions		Distance Equations
	x-axis	y-axis	
case 1	$r.x_l \le q.x \le r.x_u$	$r.y_l \le q.y \le r.y_u$	distance = 0
case 2	$q.x < r.x_l$	$r.y_l \le q.y \le r.y_u$	distance = $(q.x - r.x_l)^2$
case 3	$q.x > r.x_u$	$r.y_l \le q.y \le r.y_u$	distance = $(q.x - r.x_u)^2$
case 4	$r.x_l \le q.x \le r.x_u$	$q.y < r.y_l$	distance = $(q.y - r.y_l)^2$
case 5	$r.x_l \le q.x \le r.x_u$	$q.y > r.y_u$	distance = $(q.y - r.y_u)^2$
case 6	$q.x < r.x_l$	$q.y < r.y_l$	distance = $(q.x - r.x_l)^2 + (q.y - r.y_l)^2$
case 7	$q.x < r.x_l$	$q.y > r.y_u$	distance = $(q.x - r.x_l)^2 + (q.y - r.y_u)^2$
case 8	$q.x > r.x_u$	$q.y < r.y_l$	distance = $(q.x - r.x_u)^2 + (q.y - r.y_l)^2$
case 9	$q.x > r.x_u$	$q.y > r.y_u$	distance = $(q.x - r.x_u)^2 + (q.y - r.y_u)^2$

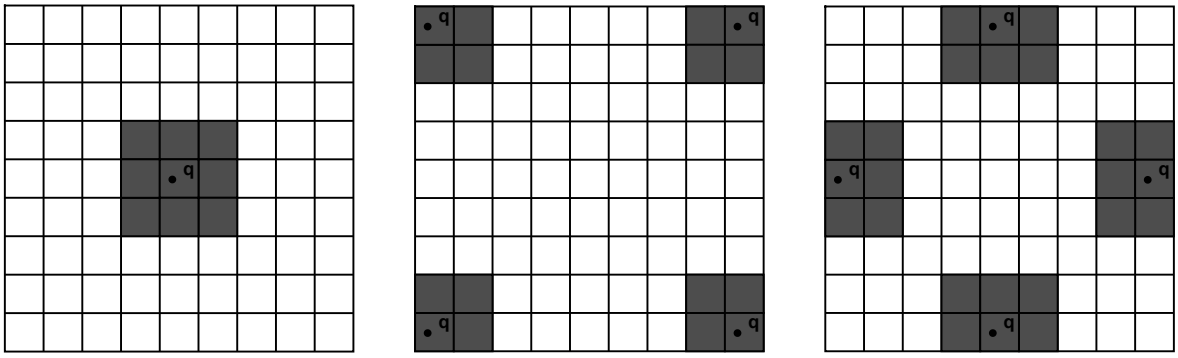
Σχήμα 3.8: Συνθήκες & εξισώσεις αποστάσεων

3.3.2 Δομές δεδομένων

Όπως στην ευρετηρίαση, έτσι και σε αυτό το στάδιο χρειαζόμαστε κάποιες δομές για την ανάπτυξη των ερωτημάτων. Επιλέγουμε λοιπόν, να χρησιμοποιήσουμε δύο σωρούς απο-

θήκευσης. Ο πρώτος αφορά την καταγραφή των αντικειμένων, που συλλέγονται ως πιθανά αποτελέσματα ενός ερωτήματος, ενώ ο δεύτερος αφορά την καταγραφή των κελιών, τα οποία πρόκειται να επισκεφθούμε.

Αρχικά, για την περίπτωση των αντικειμένων, κάνουμε χρήση ενός σωρού μεγίστου, σταθερού μεγέθους, που ισούται με το πλήθος των k αντικειμένων που αναζητούμε σε κάθε ερώτημα. Μέχρι να ξεκινήσει ένα ερώτημα, η συγκεκριμένη δομή παραμένει κενή. Για την περίπτωση των κελιών, κάνουμε χρήση ενός σωρού ελαχίστου, μεταβλητού μεγέθους. Ο λόγος του ακαθόριστου μεγέθους σχετίζεται με την άγνοια που έχουμε για το πλήθος των κελιών που πρόκειται να επισκεφθούμε κατά την εκτέλεση ενός ερωτήματος.



(α) (β) (γ)

Σχήμα 3.9: Περιπτώσεις αρχικοποίησης του σωρού των κελιών

Εδώ πρέπει να τονιστεί ότι ο σωρός των κελιών χρήζει διαφορετικής διαχείρισης από αυτή του σωρού των αντικειμένων. Συγκεκριμένα, είναι απαραίτητο να αρχικοποιηθεί με κάποια κελιά (ή τουλάχιστον με ένα), ώστε να είναι εφικτή η εκτέλεση του ερωτήματος. Στη δική μας προσέγγιση και κατά την έναρξη ενός ερωτήματος, επιλέγουμε ο σωρός των κελιών να εμπεριέχει:

- Εννιά κελιά, αν η θέση του ερωτήματος αντιστοιχεί σε κάποιο κεντρικό κελί του πλέγματος (Σχήμα 3.9α).
- Τέσσερα κελιά, αν η θέση του ερωτήματος αντιστοιχεί σε κάποιο από τα κελιά, στις γωνίες του πλέγματος (Σχήμα 3.9β).
- Έξι κελιά, αν η θέση του ερωτήματος αντιστοιχεί σε κάποιο κελί, στις άκρες του πλέγματος (Σχήμα 3.9γ).

Προφανώς, η διάταξη των κελιών εντός του σωρού, γίνεται με βάση την απόστασή τους από το σημείο του ερωτήματος, όπως ακριβώς και στη διάταξη των αντικειμένων στον αντίστοιχο σωρό τους. Καθώς τα κελιά χαρακτηρίζονται εξίσου ως πολύγωνα, οι περιπτώσεις που περιγράφονται στα Σχήματα 3.7 και 3.8, ισχύουν και για τις αποστάσεις των κελιών από το σημείο του ερωτήματος.

3.3.3 Βήματα αλγορίθμου

Μέχρι στιγμής, η δημιουργία-αρχικοποίηση των σωρών αποτελεί το πρώτο βήμα του αλγορίθμου, που επιδιώκουμε να αναπτύξουμε. Μαζί με τους σωρούς, ακολουθεί και η δήλωση μίας μεταβλητής, η οποία έχει το ρόλο ενός bound. Στην ουσία, σε αυτή τη μεταβλητή κρατάμε διαρκώς την απόσταση του αντικειμένου που βρίσκεται εντός του σωρού και χαρακτηρίζεται ως το πιο μακρινό από το σημείο μας. Μέχρι ο σωρός να γεμίσει πλήρως, το bound επιλέγεται να έχει άπειρη τιμή.

Στο επόμενο βήμα του αλγορίθμου, ξεκινάμε ένα βρόχο, στον οποίο για όσο υπάρχουν κελιά στο αντίστοιχο σωρό, αφαιρούμε κάθε φορά εκείνο με τη μικρότερη απόσταση από το σημείο του ερωτήματος. Αμέσως μετά την αφαίρεσή του, ακολουθεί ο πρώτος έλεγχος, του κατά πόσο η απόστασή του από το σημείο, δεν υπερβαίνει την τιμή του bound.

Ο συγκεκριμένος έλεγχος βασίζεται στο γεγονός ότι ένα κελί βρίσκεται πάντοτε πιο κοντά στο ερώτημα, σε σχέση με οποιοδήποτε από τα αντικείμενα που εμπεριέχει. Αν λοιπόν επιβεβαιωθεί ότι η απόστασή του είναι μεγαλύτερη του bound, το ίδιο ακριβώς θα ισχύει και για τα αντικείμενά του, οπότε δεν υπάρχει λόγος να εξετάσουμε περαιτέρω για το αν κάποιο από αυτά ανήκει στα αποτελέσματα του ερωτήματος. Επιπλέον, καθώς το εν λόγω κελί επιλέχθηκε ως το πιο κοντινό στο σημείο, είναι άσκοπο να εξετάσουμε τα υπόλοιπα κελιά του αντίστοιχου σωρού, επομένως ο αλγόριθμος τερματίζεται σε αυτό το στάδιο.

Αν τώρα επικρατήσει το ενδεχόμενο, στο οποίο το αφαιρούμενο κελί βρίσκεται πιο κοντά στο σημείο, από οτι ορίζει το bound, ακολουθούν ακόμη δύο βήματα. Στο πρώτο, αρχίζουμε να ανακτούμε ένα-ένα τα αντικείμενα του κελιού, εξετάζοντας αν η απόστασή του από το σημείο αναφοράς, το καθιστά ικανό να εισέλθει στο σωρό των αντικειμένων. Μόλις εξετασθούν όλα τα αντικείμενα, ελέγχουμε αν χρειάζεται κάποια ενημέρωση και στη τιμή του bound. Στο δεύτερο και πρακτικά τελευταίο βήμα του αλγορίθμου, έχουμε την ανάκτηση των κελιών-γειτόνων, του υπό διερεύνηση κελιού. Συγκεκριμένα, εκτελούμε ένα βρόχο, στον οποίο ελέγχουμε για κάθε γειτονικό κελί, ότι η απόσταση του δεν υπερβαίνει το bound. Εφό-

σον η συνθήκη αποδειχθεί ορθή, προχωρούμε στην ένθεσή του στο σωρό των κελιών.

Algorithm 1: Steps of k-NN searching

```
for each query point q do
    Step 1
    Create Max-Heap O
    Create and initialize Min-Heap H with the appropriate cells
    Set bound := ∞
    while H.nonempty() do
        Step 2
        c := H.pop()
        if dist(c, q) ⩾ bound then
            break                          // k-NN is confirmed
        else
            Update the Heap O using all objects in c
            Set bound := O.top.dist
        end
        Step 3
        for each neighbor n of cell c do
            Compute dist(n, q)
            if dist(n, q) < bound then
                Insert cell n in the Heap H
            end
        end
    end
end
```

Όλα τα παραπάνω βήματα που αναλύθηκαν εκτενώς, μπορούμε να τα δούμε συγκεντρωτικά στον αλγόριθμο 1. Από πλευράς κατανόησης, προσπαθούμε να είναι όσο το δυνατό πιο απλός, αποτελούμενος από περιορισμένο αριθμό συνθηκών ελέγχου, και από ελάχιστα βήματα, με ξεκάθαρο στόχο του τι κάνει το καθένα από αυτά. Από πλευράς υλοποίησης ωστόσο, εμφανίζει κάποια γκρίζα σημεία, τα οποία χρειάζονται ιδιαίτερη αντιμετώπιση, για να είναι επιτυχημένο το αποτέλεσμα, που προκύπτει από την εφαρμογή του.

Τα γκρίζα αυτά σημεία σχετίζονται αποκλειστικά με τις διπλότυπες εμφανίσεις τόσο των αντικειμένων, όσο και των κελιών, στους σωρούς που χρησιμοποιούμε. Σε όλη την υπόλοιπη ενότητα, θα ασχοληθούμε μόνο με την εξάλειψη αυτών των διπλότυπων, παραθέτοντας ορι-

σμένες μεθόδους, οι οποίες δοκιμάστηκαν για την αντιμετώπιση αυτού του ζητήματος.

3.3.4 Διπλότυπες εμφανίσεις στο σωρό των αντικειμένων

Μέχρι στιγμής, με τη διαμέριση και την ευρετηρίαση που προτείνουμε, κατορθώνουμε να διαχειριστούμε τα αντικείμενα που εκτείνονται σε πολλαπλά κελιά, αναθέτοντας ένα αντίγραφό τους σε κάθε εμπλεκόμενο κελί. Μπορεί με αυτή την προσέγγιση να γλιτώνουμε αρκετούς ελέγχους όσον αφορά την εύρεση του κελιού που ανήκει το κάθε αντικείμενο, ωστόσο δεν εξαλείφει το ενδεχόμενο των διπλότυπων εμφανίσεων στο σύνολο των αποτελεσμάτων. Εδώ λοιπόν είναι το στάδιο, στο οποίο αρχίζει να αποκτά ιδιαίτερη σημασία ο διαχωρισμός των αντικειμένων ενός κελιού, σε κλάσεις.

Ας υποθέσουμε ότι αρχίζουμε την εκτέλεση ενός ερωτήματος, το οποίο τοποθετείται σε κάποιο κεντρικό κελί του πλέγματος. Ο σωρός των κελιών μας αρχικοποιείται με εννιά κελιά, εμπεριέχοντας το κελί του ερωτήματος, το οποίο χαρακτηρίζουμε ως Cq, και τα οχτώ κελιά γύρω από αυτό. Βάσει του αλγορίθμου, το πρώτο κελί που εξετάζεται είναι το Cq, λόγω της μηδενικής απόστασης που έχει από το σημείο του ερωτήματος. Στην περίπτωσή του και καθώς φτάνουμε στο βήμα ενημέρωσης του σωρού των αντικειμένων, επισκεπτόμαστε και τις δεκαέξι κλάσεις του, εξετάζοντας όλα τα αντικείμενά του.

Αμέσως μετά, βάσει των αποστάσεων, ακολουθούν τα κελιά που βρίσκονται στο νοητό σταυρό, με κέντρο το κελί Cq. Για αυτά τα κελιά, λαμβάνοντας υπόψιν τα αντικείμενα που ήδη έχουν εξεταστεί από την περίπτωση του Cq, παρατηρούμε ότι υπάρχουν σύνολα από οχτώ κλάσεις - διαφορετικό για καθένα από τα τέσσερα κελιά - τα οποία παραπέμπουν σε αντικείμενα, τα οποία δεν έχουν ακόμη εξεταστεί. Κάτι ανάλογο συναντάται και στα κελιά που βρίσκονται διαγώνια του κελιού Cq. Σε αυτό το σημείο ωστόσο, λαμβάνουμε υπόψιν μας και τα αντικείμενα που εξετάζονται από τα κελιά του σταυρού. Επομένως, καταλήγουμε πάλι σε σύνολα κλάσεων, αποτελούμενα αυτή τη φορά, από τέσσερις κλάσεις.

Για την περίπτωση αυτών των εννιά κελιών, βλέπουμε ότι υπάρχει τρόπος να αποφύγουμε τις διπλότυπες εμφανίσεις των αντικειμένων, βασίζοντας την ανάκτησή τους σε συγκεκριμένα σύνολα κλάσεων. Το ζητούμενο ωστόσο είναι, να διαχειριζόμαστε τα διπλότυπα για κάθε κελί του χώρου. Συνεχίζουμε λοιπόν την εξέταση επιπλέον κελιών, από τα επόμενα επίπεδα του πλέγματος, έχοντας ως δεδομένο ότι τα κελιά που ανήκουν στο σταυρό, έχουν μικρότερη απόσταση σε σχέση με τα κελιά, στις διαγωνίους.

Από αυτή τη διαδικασία, καταλήγουμε τελικά στο συμπέρασμα ότι τα σύνολα των εννιά

πρώτων κελιών, εξακολουθούν να εμφανίζονται και στις περιπτώσεις άλλων κελιών. Συγκεκριμένα, παρατηρούμε ότι στα κελιά του σταυρού αντιστοιχεί πάντοτε μία οχτάδα κλάσεων, με αντικείμενα που δεν έχουν εξεταστεί, ενώ στα διαγώνια κελιά μία τετράδα κλάσεων αντίστοιχα. Μάλιστα, η κάθε οχτάδα ή τετράδα κλάσεων, καταλήγει να είναι κοινή για όσα κελιά εκτείνονται προς την ίδια κατεύθυνση του χώρου, επομένως τα εν λόγω σύνολα μετεξελίσσονται σε μοτίβα κλάσεων.

Patterns	Conditions	Classes
Pattern 0	c.x < Cq.x & c.y < Cq.y	{A, C, I, K}
Pattern 1	c.x = Cq.x & c.y < Cq.y	{A, C, E, G, I, K, M, O}
Pattern 2	c.x > Cq.x & c.y < Cq.y	{A, C, E, G}
Pattern 3	c.x < Cq.x & c.y = Cq.y	{A, B, C, D, I, J, K, L}
Pattern 4	Only cell Cq	ALL CLASSES
Pattern 5	c.x > Cq.x & c.y = Cq.y	{A, B, C, D, E, F, G, H}
Pattern 6	c.x < Cq.x & c.y > Cq.y	{A, B, I, J}
Pattern 7	c.x = Cq.x & c.y > Cq.y	{A, B, E, F, I, J, M, N}
Pattern 8	c.x > Cq.x & c.y > Cq.y	{A, B, E, F}

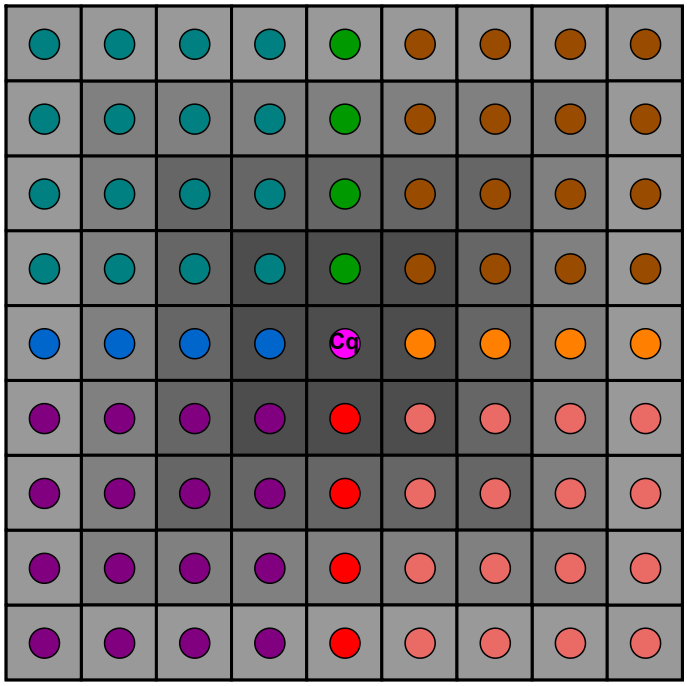
Σχήμα 3.10: Μοτίβα κλάσεων

Στο Σχήμα 3.10, παρουσιάζουμε όλα τα μοτίβα, δείχνοντας το ακριβές σύνολο κλάσεων, που αντιστοιχίζεται σε κάθε περίπτωση. Επιπλέον, συμπεριλαμβάνουμε κάποιες σχέσεις που εκφράζουν μαθηματικά, την κατεύθυνση των κελιών ανά μοτίβο. Στην ουσία, για αυτές τις σχέσεις επικαλούμαστε τα σημεία που χαρακτηρίζουν τα κέντρα των κελιών, συσχετίζοντας τη θέση τους, με το αντίστοιχο σημείο του κελιού Cq.

Με τον εντοπισμό των μοτίβων, ο χώρος αποκτά μία μορφή, όπως αυτή απεικονίζεται στο Σχήμα 3.11. Προφανώς, τα αντίστοιχα μοτίβα κλάσεων ισχύουν και για τις κατηγορίες όπου το ερώτημα ανήκει σε κάποιο κελί, στις γωνίες ή στις άκρες του πλέγματος. Σε αυτές ωστόσο, κάνουν την εμφάνιση τους σχεδόν οι μισές από τις περιπτώσεις των μοτίβων. Αυτός είναι λοιπόν ο λόγος που προτιμήθηκε η κατηγορία, στην οποία το ερώτημα εντοπίζεται σε κάποιο κεντρικό κελί, ώστε να είναι πιο πλήρης και περιγραφική η ανάλυσή μας.

Συνοψίζοντας όσα είδαμε σε αυτή την υποενότητα, είναι αντιληπτό πως τα μοτίβα κλά-

σεων λειτουργούν αποτελεσματικά στην προσπάθειά μας να αποφύγουμε τις επαναλαμβανόμενες εμφανίσεις αντικειμένων, στα αποτελέσματα ενός ερωτήματος. Επίσης, η εύρεση του μοτίβου, που αντιστοιχεί στην περίπτωση του κάθε κελιού, μπορεί να επιτευχθεί με χαμηλό υπολογιστικό κόστος. Για αυτό λόγο, επιλέγουμε κατά την ένθεση των κελιών στο σωρό, να συμπεριλάβουμε, μαζί με τα αναγνωριστικά και τις αποστάσεις τους, αυτή την επιπλέον πληροφορία.



Σχήμα 3.11: Αναπαράσταση του χώρου με βάση τα μοτίβα κλάσεων

3.3.5 Διπλότυπες εμφανίσεις στο σωρό των κελιών

Μετά τη διαχείριση των διπλότυπων αντικειμένων, σειρά έχει η διαχείριση του ίδιου ζητήματος, αυτή τη φορά στο σωρό των κελιών. Σύμφωνα με τον αλγόριθμό μας, κάθε φορά που εξετάζουμε ένα κελί, το τελευταίο βήμα που εφαρμόζεται, είναι ο έλεγχος σχετικά με το ποια γειτονικά κελιά του, πληρούν τις προϋποθέσεις, για να εισέλθουν στο σωρό. Αφήνοντας λοιπόν ανεξέλεγκτη την ένθεση των κελιών, είναι σίγουρο ότι τα περισσότερα από αυτά, θα καταλήξουν να εμφανίζονται περισσότερες από μία φορές.

Το ενδεχόμενο πολλαπλών εμφανίσεων ενός κελιού στο σωρό, πέρα του ότι δεν επιθυμητή από πλευράς επίδοσης του αλγορίθμου, αλλοιώνει και τα αποτελέσματα ενός ερωτήματος. Για να γίνει κατανοητό, ας υποθέσουμε ότι έχουμε ένα κελί c, το οποίο εντοπίζεται πολλαπλές φορές στο σωρό των κελιών, ως γειτονικό, σε κελιά τα οποία ήδη έχουν εξετα-

στεί. Επιπλέον, ας θεωρήσουμε ότι στο κελί c εμπεριέχονται ορισμένα αντικείμενα, τα οποία ανήκουν στα τελικά αποτελέσματα ενός ερωτήματος.

Όταν έρθει η στιγμή που το κελί c χαρακτηρίζεται ως το κοντινότερο εντός του σωρού, στην ουσία θα γίνει συνεχόμενα η εξέταση όλων των εμφανίσεών του. Το αποτέλεσμα αυτής της εξέλιξης είναι, ο σωρός των αντικειμένων να γεμίσει με αντικείμενα, τα οποία επαναλαμβάνονται και δεν πρόκειται να αντικατασταθούν στην πορεία εκτέλεσης του ερωτήματος, καθώς όλα αναγνωρίζονται ως αποτελέσματά του.

Με την πιθανότητα εμφάνισης ενός τέτοιου σεναρίου εκτέλεσης, ενισχύεται ακόμα περισσότερο η ανάγκη για αντιμετώπιση των διπλότυπων κελιών. Έτσι λοιπόν, έχουμε αναπτύξει τέσσερις μεθόδους, οι οποίες διαχειρίζονται αυτό το ζήτημα. Στο υπόλοιπο αυτής της υποενότητας θα παρουσιάσουμε μία προς μία αυτές τις μεθόδους, εξηγώντας εκτενώς τη λογική πίσω από την ανάπτυξή τους.

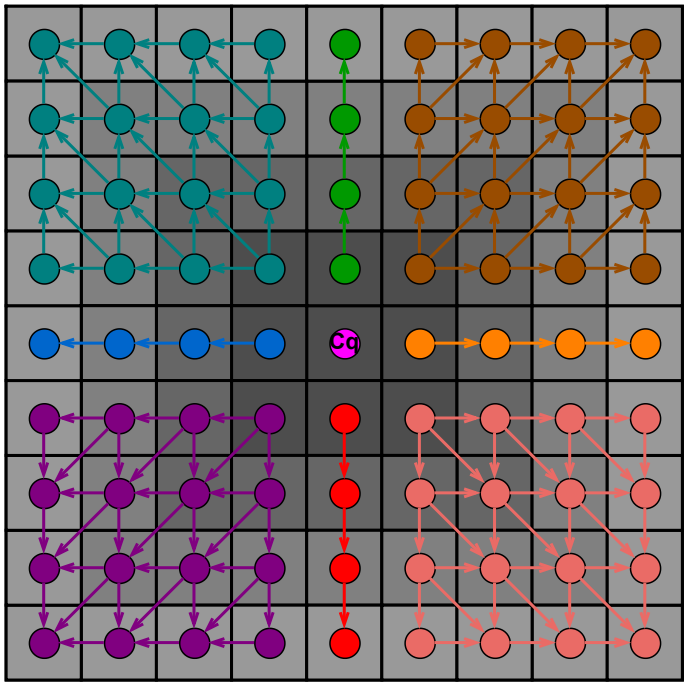
3.3.5.1 Μέθοδος 1^{η}

Η πρώτη μας απόπειρα για την αποφυγή διπλότυπων εμφανίσεων στο σωρό των κελιών, βασίζεται στη λογική καταγραφής όσων κελιών επισκεπτόμαστε. Συγκεκριμένα, επιλέγουμε να χρησιμοποιήσουμε μία λίστα, στην οποία, για κάθε κελί που εξετάζεται η δυνατότητα του να εισέλθει στον αντίστοιχο σωρό, καταγράφεται το αναγνωριστικό του.

Πάνω σε αυτή τη λογική, εισάγουμε έναν επιπλέον έλεγχο στο στάδιο αναζήτησης γειτονικών κελιών. Πλέον για κάθε γειτονικό κελί, ελέγχουμε σε πρώτη φάση αν έχει ήδη διαπεραστεί, από προγενέστερο βήμα του αλγορίθμου, ψάχνοντας το αναγνωριστικό του στη λίστα επισκέψεων. Από τη μία, αν το αναγνωριστικό δεν βρεθεί, πραγματοποιούμε την ένθεσή του στη λίστα, και ελέγχουμε σε δεύτερη φάση για το αν ικανοποιείται η συνθήκη, που θέλει η απόσταση του κελιού να μην υπερβαίνει το bound. Από τη συνθήκη αυτή και μετά, η διαδικασία παραμένει ακριβώς όπως την περιγράψαμε στην υποενότητα 3.3.3. Από την άλλη, αν το αναγνωριστικό εντοπιστεί στη λίστα, παρακάμπτουμε τον επόμενο έλεγχο και προχωράμε στο επόμενο γειτονικό κελί που μάς ενδιαφέρει.

Το μόνο που δεν έχει αναφερθεί, είναι το πόσα γειτονικά κελιά ελέγχουμε σε κάθε βήμα του αλγορίθμου. Εδώ λοιπόν, υπάρχουν δύο κατηγορίες. Στην πρώτη, η οποία έχει να κάνει με τα κελιά του σταυρού, επιλέγουμε κάθε φορά να εξετάζουμε ένα γείτονά τους, αυτόν που βρίσκεται στην ίδια κατεύθυνση και στο επόμενο επίπεδο κελιών του χώρου. Στη δεύτερη κατηγορία, επιλέγουμε να εξετάζουμε κάθε φορά τους τρεις γείτονες ενός κελιού, προς στις

κατευθύνσεις που υποδεικνύουν τα βέλη, στο Σχήμα 3.12. Από αυτή την απεικόνιση, είναι προφανές, ότι η λίστα επισκέψεων, θεωρείται περιττή, για τα κελιά του σταυρού.



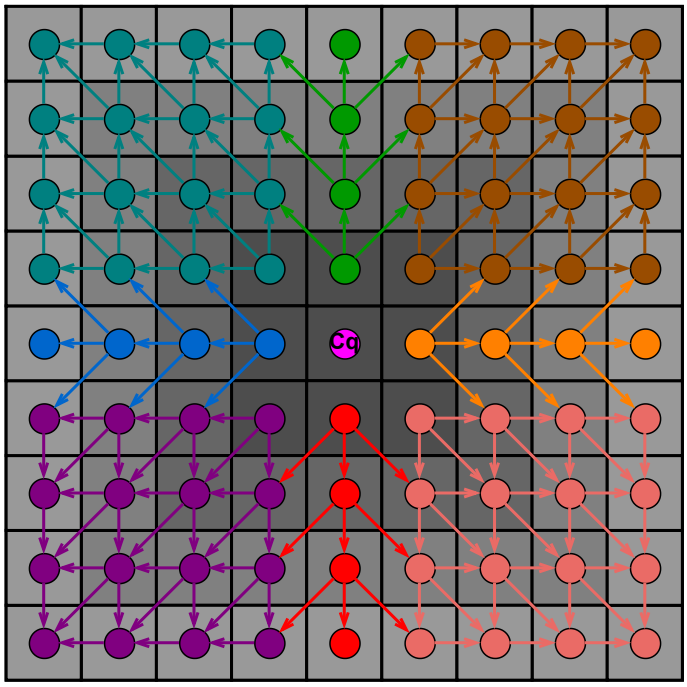
Σχήμα 3.12: Αναζήτηση γειτονικών κελιών με βάση τη πρώτη μέθοδο

3.3.5.2 Μέθοδος 2^{η}

Η μέθοδος αυτή, πρόκειται στην ουσία για παραλλαγή της προηγούμενης. Η διαφορά τους σχετίζεται με το πλήθος γειτονικών κελιών που εξετάζονται, στην περίπτωση του σταυρού. Όπως φαίνεται και στο Σχήμα 3.13, αυτή τη φορά επιλέγουμε, μαζί με το κελί του επόμενου επιπέδου, να ελέγχουμε επίσης τα κελιά που βρίσκονται αριστερά και δεξιά του (ή πάνω και κάτω αντίστοιχα). Πλέον, η λίστα επισκέψεων γίνεται απαραίτητη για τα επιπλέον γειτονικά κελιά του σταυρού, καθώς χαρακτηρίζονται ως γείτονες και άλλων κελιών του χώρου.

Αν και η συγκεκριμένη μέθοδος δεν διαφοροποιείται αρκετά από την πρώτη, υπάρχουν λόγοι που μάς οδήγησαν στην επιλογή της. Καταρχάς, με την αύξηση των γειτονικών κελιών στην περίπτωση του σταυρού, κατορθώνουμε να υπάρχει μία ομοιομορφία, σε όλες τις επαναλήψεις της διεκπεραίωσης ενός ερωτήματος. Κατά δεύτερον, αυξάνουμε την πιθανότητα εντοπισμού των κελιών με τις μικρότερες αποστάσεις, σε όσο το δυνατόν λιγότερα βήματα εκτέλεσης. Αυτή η ενέργεια δεν μειώνει τις επαναλήψεις που χρειάζονται για να ολοκληρωθεί ένα ερώτημα. Κυρίως βοηθάει στο να είναι ιεραρχική η ένθεση των κελιών στο σωρό,

χωρίς να χρειάζονται διαρκώς αναδιατάξεις στο εσωτερικό του.



Σχήμα 3.13: Αναζήτηση γειτονικών κελιών με βάση τη δεύτερη μέθοδο

3.3.5.3 Μέθοδος 3^{η}

Με τις δύο πρώτες μεθόδους, είναι γεγονός ότι το ζήτημα των διπλότυπων κελιών, αντιμετωπίζεται επιτυχημένα. Πέρα όμως από την αποτελεσματικότητα, μάς απασχολεί ιδιαίτερα και ο παράγοντας της αποδοτικότητας, στον οποίο ωστόσο, υστερούν και οι δύο. Κύρια αιτία της εξέλιξης αυτής, αποτελεί ο τρόπος με τον οποίο αναζητούμε τα γειτονικά κελιά, σε συνδυασμό με τη χρήση της λίστας επισκέψεων.

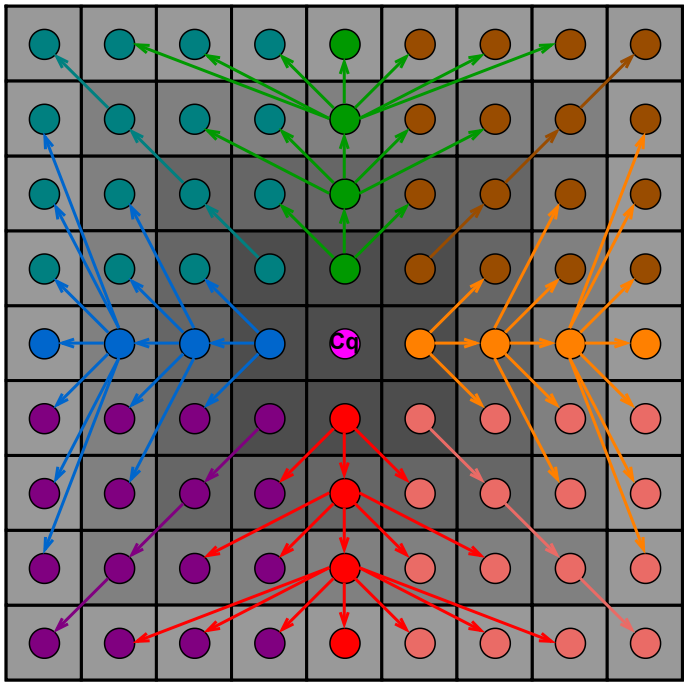
Από τα Σχήματα 3.12 και 3.13, φαίνεται πως όλα τα κελιά του χώρου (εκτός από εκείνα του σταυρού, στην πρώτη μέθοδο), εντοπίζονται ως γειτονικά, σε περισσότερα από ένα βήματα του αλγορίθμου. Κατά συνέπεια, αν εξαιρέσουμε την πρώτη φορά που τα διαπερνάμε, στην οποία γίνεται η καταγραφή τους λίστα επισκέψεων και αποφασίζεται αν μπορούν να εισέλθουν στο σωρό, όλες οι υπόλοιπες επισκέψεις μας καταλήγουν εν τέλει σε περιττούς ελέγχους.

Αφήνουμε λοιπόν στην άκρη, τόσο το τωρινό τρόπο διαπέρασης των γειτονικών κελιών, όσο και τη λογική της καταγραφής τους, επικεντρώνοντας το ενδιαφέρον μας σε μία νέα μέθοδο. Αυτή τη φορά, επιλέγουμε η αναζήτηση των γειτόνων να εφαρμόζεται πάντα στο επόμενο επίπεδο του πλέγματος, φροντίζοντας το κάθε κελί να εξετάζεται αποκλειστικά σε

ένα βήμα του αλγορίθμου. Και σε αυτή τη μέθοδο, διαχωρίζουμε τα κελιά ανάλογα με το πλήθος γειτόνων που αναζητούν, διακρίνοντας τρεις κατηγορίες: (i) τα κελιά στο σταυρό, (ii) τα κελιά στα άκρα κάθε επιπέδου και (iii) τα κελιά που βρίσκονται ανάμεσα τους

Για τα κελιά του σταυρού, αποφασίζουμε ο αριθμός των γειτόνων να είναι δυναμικός, εξαρτώμενος από το βαθμό του επιπέδου, στον οποίο εντοπίζονται. Έτσι λοιπόν για τα κελιά του επιπέδου 0 αναζητούμε τρεις γείτονες από το επίπεδο 1, για τα κελιά του επιπέδου 1, πέντε γείτονες από το επίπεδο 2 κ.ο.κ. Με λίγα λόγια, αναζητούμε κάθε φορά $2 \times level + 1$ γείτονες (όπου level, το επίπεδο που ανήκουν), στους οποίους περιλαμβάνονται το κεντρικό κελί, μαζί με τα level δεξιά και τα level αριστερά κελιά του (ή τα level πάνω και τα level κάτω αντίστοιχα).

Για τα δεύτερη κατηγορία και τα κελιά στις άκρες, επιλέγουμε να αναζητούμε ένα μόνο γείτονα, και συγκεκριμένα εκείνον που εντοπίζεται διαγώνιά τους, σε άκρο το επόμενου επιπέδου. Τέλος, για τα ενδιάμεσα κελιά, το στάδιο αναζήτησης γειτονικών κελιών παραλείπεται. Η απόφαση αυτή μπορεί να δικαιολογηθεί από το Σχήμα 3.14. Σε αυτό μπορούμε να δούμε πως οι δύο πρώτες κατηγορίες κελιών αρκούν, για να επιτευχθεί το ζητούμενο της αποκλειστικής διαπέρασης όλων των γειτόνων.



Σχήμα 3.14: Αναζήτηση γειτονικών κελιών με βάση τη τρίτη μέθοδο

3.3.5.4 Μέθοδος $4^η$

Μέχρι στιγμής, η τρίτη μέθοδος είναι η καλύτερη προσέγγιση που έχουμε παρουσιάσει. Παρά την υπεροχή της από τις προηγούμενες, ο τρόπος ανάπτυξής της, επηρεάζει σε κάποιο βαθμό την επίδοση του αλγορίθμου. Στην ουσία, όσο ανοιγόμαστε στο χώρο και διαπερνάμε κελιά από τα πιο μακρινά επίπεδα του πλέγματος, αυξάνεται και ο αριθμός των κελιών που εξετάζουμε. Από τη μία, αυτή η διαδικασία μάς εξασφαλίζει τη δυνατότητα να μπορούμε να επισκεφθούμε οποιοδήποτε κελί του χώρου. Από την άλλη όμως, έχει υψηλό κόστος, καθώς για κάθε κελί χρειάζεται να πραγματοποιήσουμε έναν υπολογισμό (αυτόν της απόστασής του) και έναν έλεγχο (αν η απόσταση υπερβαίνει το bound).

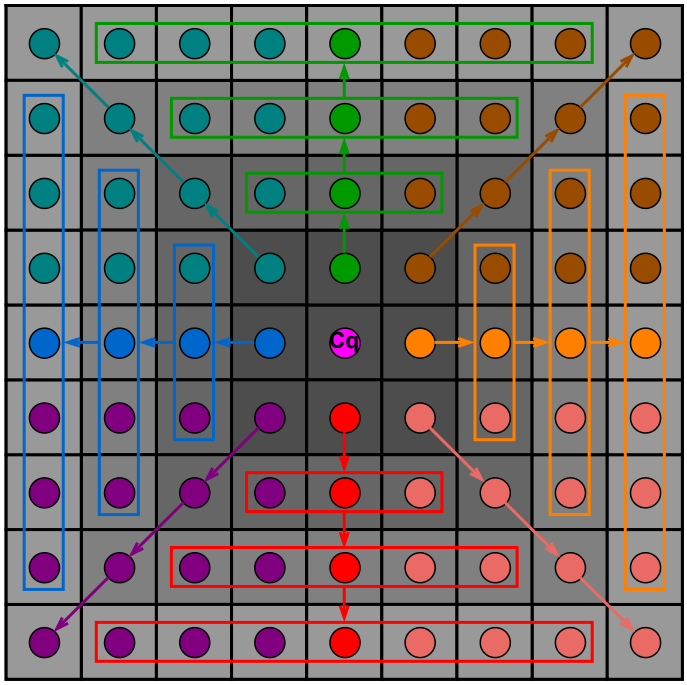
Δεδομένου ότι το bound έχει την τάση να μειώνεται, στην πορεία εκτέλεσης του αλγορίθμου, το να επισκεπτόμαστε κελιά, όλο και πιο απομακρυσμένα από το σημείο του ερωτήματος, κρύβει τον κίνδυνο των άσκοπων υπολογισμών και ελέγχων. Το πρόβλημα δεν αφορά μόνο τα κελιά που δεν κατορθώνουν να εισέλθουν στο σωρό. Ακόμη και αυτά που ικανοποιούν τη συνθήκη και εισέρχονται, δεν είναι σίγουρο ότι μέχρι την ολοκλήρωση του ερωτήματος, θα έχουν ανακτηθεί. Συνεπώς, όσο πιο γεμάτος καταλήγει να είναι ο σωρός στο τέλος της εκτέλεσης, τόσο πιο μεγάλο είναι και το κόστος.

Με βάση τα όσα αναλύσαμε, επιλέγουμε να αναπτύξουμε ακόμη μία μέθοδο, η οποία κρατάει τη λογική των επιπέδων, περιορίζοντας ωστόσο το επιβαρυντικό κόστος. Αυτή τη φορά, όταν εντοπίζουμε μία γειτονιά κελιών, αποφασίζουμε να μην κάνουμε απευθείας τους ελέγχους που δείχνουν ποια είναι ικανά να εισέλθουν στο σωρό, αλλά προτιμάμε να καταγράψουμε το ορθογώνιο που τα περικλείει. Με τον όρο ορθογώνιο, εννοείται μία δομή, ίδιου τύπου με αυτή των κελιών, στην οποία εμπεριέχεται όλη η απαραίτητη πληροφορία των εμπλεκόμενων κελιών (αναγνωριστικά και μοτίβα κλάσεων).

Για την αποθήκευση αυτών των ορθογώνιων, επικαλούμαστε την απόσταση του κελιού που βρίσκεται πιο κοντά στο ερώτημα, το οποίο δεν είναι άλλο από το κελί που ανήκει στο σταυρό. Πλέον, όταν κάνουμε μία εξαγωγή από το σωρό, ελέγχουμε αν πρόκειται για κελί ή ορθογώνιο. Στην περίπτωση κελιού, η διαδικασία παραμένει όπως την έχουμε παρουσιάσει. Στην περίπτωση ορθογώνιου, προχωράμε με την εξέταση των κελιών που εμπεριέχει, και μετέπειτα με την εισαγωγή τους στο σωρό. Πρακτικά, η αποθήκευση ενός ορθογώνιου, που φέρει την πληροφορία μίας γειτονιάς κελιών, λειτουργεί ως ενδιάμεσο βήμα.

Προς το παρόν, ακολουθούμε μία μικτή προσέγγιση στην αναζήτηση των γειτονικών κελιών. Όπως φαίνεται και στο Σχήμα 3.15, για τα κελιά που βρίσκονται στις άκρες, εξακολου-

θούμε να εργαζόμαστε ακριβώς όπως και στη τρίτη μέθοδό μας. Για να υπάρξει ομοιομορφία στο τρόπο που επισκεπτόμαστε τα γειτονικά κελιά, βασιζόμαστε σε μία νέα προσέγγιση, η οποία έχει προταθεί σε παρεμφερή ερευνητική εργασία διεκπεραίωσης ερωτημάτων πλησιέστερων γειτόνων [26].



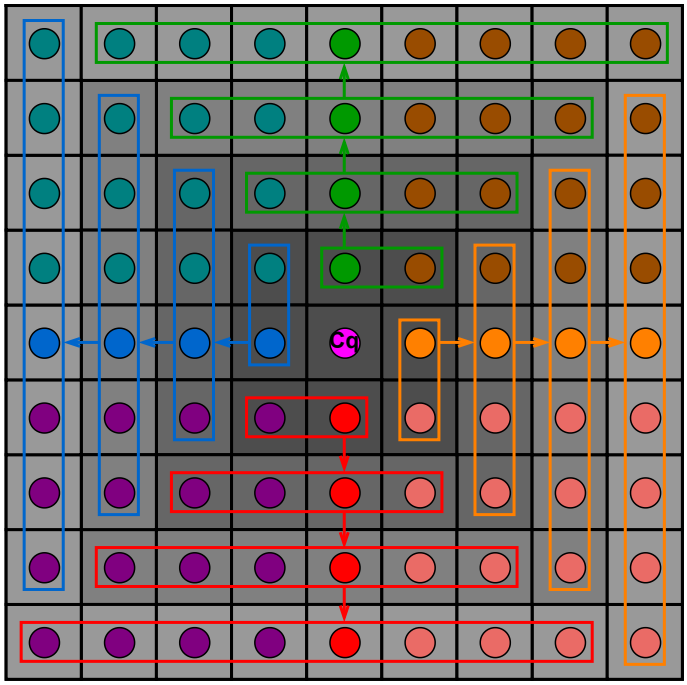
Σχήμα 3.15: Αναζήτηση γειτονικών κελιών με βάση τη τέταρτη μέθοδο

Στη καινούργια λοιπόν προσέγγιση, ξεκινάμε την εφαρμογή των ορθογώνιων από το βήμα αρχικοποίησης του σωρού. Πλέον, για όλα τα κελιά εκτός το κεντρικού Cq, προχωράμε στην καταγραφή των πληροφοριών τους, σε ορθογώνια χωρητικότητας δύο κελιών, τα οποία σχηματίζονται, ακολουθώντας τη φορά του ρολογιού. Έτσι σε κάθε ορθογώνιο, εμφανίζεται και ένα κελί του σταυρού, με την απόσταση του οποίου πραγματοποιείται και η ένθεση στο σωρό. Από το σημείο αυτό και μετά, η υλοποίηση παραμένει σχεδόν ίδια, με μόνο δύο απλές διαφοροποιήσεις.

Η πρώτη, σχετίζεται με το πλήθος των κελιών που αναζητούμε ως γείτονες. Όπως ήδη αναφέραμε, στο επίπεδο 0 αποθηκεύουμε την πληροφορία των κελιών σε ορθογώνια χωρητικότητας, ίση με δύο. Ακολουθούμε την ίδια λογική και στα επόμενα επίπεδα, αναζητώντας πάντοτε ορθογώνια που περικλείουν πλήθος κελιών, ίσο με το αμέσως επόμενο πολλαπλάσιο του δύο. Έτσι λοιπόν, στο επίπεδο 1 προκύπτουν ορθογώνια τεσσάρων κελιών, στο επίπεδο 2 ορθογώνια έξι κελιών κ.ο.κ. Με την αλλαγή στον αριθμό των γειτονικών κελιών, συμπεριλαμβάνονται πλέον και τα κελιά στις άκρες του κάθε επιπέδου, το οποίο μπορούμε να δούμε

στην αναπαράσταση του Σχήματος 3.16.

Η δεύτερη διαφοροποίηση, εμφανίζεται πάλι στο σημείο εύρεσης των γειτόνων και σχετίζεται με το πλήθος των ελέγχων-ενθέσεων που γίνονται. Αυτή τη φορά, μαζί με τον έλεγχο και την εισαγωγή (εφόσον είναι εφικτή) των κελιών του υπό διερεύνηση ορθογώνιου, επιδιώκεται η εισαγωγή και του ορθογώνιου, που αντιπροσωπεύει τη γειτονιά κελιών του επόμενου επιπέδου.



Σχήμα 3.16: Παραλλαγή τέταρτης μεθόδου

Με βάση τα όσα εξηγήσαμε, η προσέγγιση των ορθογώνιων μπορεί να αντιμετωπίσει αποτελεσματικά το ζήτημα των περιττών ελέγχων και υπολογισμών. Στην περίπτωση που μία γειτονιά κελιών δεν ανακτάται ποτέ από το σωρό, με τα ορθογώνια χρειαζόμαστε έναν έλεγχο και έναν υπολογισμό απόστασης, γλιτώνοντας τους επιπλέον $2 \times level$ ελέγχους και $2 \times level$ υπολογισμούς που πραγματοποιούνται, σύμφωνα με την τρίτη μέθοδο.

Πέρα από τον περιορισμό του κόστους, η συγκεκριμένη μέθοδος παρουσιάζει επίσης κάποια επιθυμητά στοιχεία, όπως:

- Προσφέρει, σε όλη την πορεία εκτέλεσης του αλγόριθμου, μία ομοιομορφία στον τρόπο με τον οποίο αναζητούνται τα γειτονικά κελιά.
- Διασφαλίζει ότι κάθε κελί αναζητείται και ελέγχεται μία και μοναδική φορά σε όλη τη διάρκεια της εκτέλεσης.

- Επιδιώκει την εισαγωγή των κελιών στο σωρό, με ιεραρχικό τρόπο, ώστε να χρειάζονται ελάχιστες αναδιατάξεις στο εσωτερικό του.

Στο επόμενο κεφάλαιο θα δούμε, ότι η προτίμησή μας για την εν λόγω μέθοδο, επιβεβαιώνεται και σε πρακτικό επίπεδο, μέσα από τους χρόνους εκτέλεσης του αλγόριθμου, από τους οποίους γίνεται ορατή η υπεροχή της, συγκριτικά με όλες τις υπόλοιπες απόπειρές μας.

Κεφάλαιο 4

Πειραματική Αξιολόγηση

Στο κεφάλαιο αυτό, ασχολούμαστε με την αξιολόγηση της υλοποίησής μας, όσον αφορά την επίδοσή της. Πιο αναλυτικά, στην ενότητα 4.1 περιγράφουμε εκτενώς το setup που ακολουθούμε για την ανάπτυξη της πειραματικής μας μελέτης. Στην ενότητα 4.2 παρουσιάζουμε τους χρόνους που προκύπτουν για κθεμία από τις εκδόσεις που έχουν υλοποιηθεί. Τέλος, στην ενότητα 4.3 αξιολογούμε τα αποτελέσματά μας, συγκρίνοντάς τα με τα αντίστοιχα που προκύπτουν από ήδη υπάρχουσα προσέγγιση.

4.1 Οργάνωση Πειραμάτων

4.1.1 Σύνολα αντικειμένων-Σύνολα ερωτημάτων

Για την ορθή αξιολόγηση της πρότασής μας, αποφασίζουμε να αναζητήσουμε δεδομένα, τα οποία παραπέμπουν σε αντικείμενα ρεαλιστικού χαρακτήρα. Έτσι λοιπόν, καταφεύγουμε στη χρήση συνόλων αντικειμένων, τα οποία συνδέονται με το ηπειρωτικό τμήμα των ΗΠΑ [27, 28, 16]. Από τα διαθέσιμα σύνολα της εν λόγω πηγής, επιλέγουμε να αξιοποιήσουμε τέσσερα αρχεία, τα οποία είναι και τα πιο αντιπροσωπευτικά, για να έχουμε στη μελέτη μας μια ποικιλία σχετικά με το πλήθος των αντικειμένων, καθώς και με τη μέση έκταση που καταλαμβάνουν στο χώρο. Στον Πίνακα 4.1 παρουσιάζονται συγκεντρωτικά, οι πληροφορίες που έχουμε στη διάθεσή μας, για καθένα από αυτά τα σύνολα.

Σε όλα τα αρχεία, το περιεχόμενό τους παραπέμπει σε συντεταγμένες που περιγράφουν τις πραγματικές εκτάσεις των αντικειμένων. Για να είναι λοιπόν δυνατή η εφαρμογή των δεδομένων αυτών στο filter step των ερωτημάτων (που είναι και το ζητούμενο), επεξεργαζόμαστε τις συντεταγμένες, που έχουμε στην κατοχή μας, και τις μετατρέπουμε κατάλληλα,

ώστε να χαρακτηρίζουν εν τέλει τα MBRs των αντικειμένων.

Πέρα από τα προαναφερθέντα αρχεία, κάνουμε χρήση ενός επιπλέον, το οποίο περιλαμβάνει δέκα χιλιάδες εγγραφές ερωτημάτων. Κατά την εκτέλεση του προγράμματος, κανονικοποιούμε τις συντεταγμένες των ερωτημάτων, με βάση τις μέγιστες και ελάχιστες τιμές που εντοπίζονται στο εκάστοτε σύνολο αντικειμένων. Συνεπώς, δοκιμάζεται κάθε φορά διαφορετικό σύνολο, ανάλογα με τα αντικείμενα που δίνονται σαν είσοδος.

Datasets	**Description**	**Type**	**Records**	**Avg. x-extent**	**Avg. y-extent**
T1	Primary Roads	Linestrings	12K	0.00239563	0.00376759
T2	Area Hydrography	Polygons	2.3M	3.85662e-05	7.25618e-05
T3	Linear Hydrography	Linestrings	5M	0.000142534	0.000267871
T4	Roads	Linestrings	16.8M	8.14567e-05	0.000148423

Πίνακας 4.1: Πληροφορίες για τα σύνολα των αντικειμένων

4.1.2 Παράμετροι Εκτέλεσης-Χαρακτηριστικά Συστήματος

Εκτός από τα αρχεία, υπάρχουν επιπλέον δύο παράμετροι, οι οποίες εμπλέκονται με τη σειρά τους στα αποτελέσματα της αξιολόγησής μας. Η πρώτη, έχει να κάνει με το βαθμό που διαμερίζεται ο χώρος. Στην ουσία, δοκιμάζουμε διάφορα μεγέθη διαμέρισης, αναζητώντας εκείνο που προσφέρει την πιο αποτελεσματική ευρετηρίαση. Καθώς ο τρόπος, με τον οποίο καταγράφονται τα αντικείμενα, επηρεάζει τη μετέπειτα διεκπεραίωση των ερωτημάτων, στόχος μας είναι να βρεθεί εκείνο το μέγεθος πλέγματος, που προσφέρει στο σύνολο, την καλύτερη δυνατή επίδοση.

Η δεύτερη παράμετρος, σχετίζεται με το πλήθος των πλησιέστερων γειτόνων που αναζητούνται. Γενικά, σε τέτοιου είδους ερωτήματα, η απόπειρα αναζήτησης πολύ μεγάλου αριθμού αντικειμένων, δεν εμπίπτει στην κατηγορία των ρεαλιστικών σεναρίων εκτέλεσης. Επίσης, με βάση τα δεδομένα που έχουμε στη διάθεσή μας, και η αναζήτηση πολύ μικρού αριθμού (π.χ. δύο αντικείμενα), δεν έχει ιδιαίτερη σημασία. Έτσι λοιπόν, αποφασίζουμε για καθένα από τα σύνολα αντικειμένων, να ασχοληθούμε αποκλειστικά με τρεις τιμές k, αναπτύσσοντας εν τέλει όλα τα πειράματά μας για την αναζήτηση 10, 30 και 50 αντικειμένων.

Στον Πίνακα 4.2, συνοψίζονται όλες οι παράμετροι που σχετίζονται με την εκτέλεση του προγράμματος. Στο σημείο αυτό αξίζει να κάνουμε μία επισήμανση σχετικά με τις παραμέ-

τρους της μεταγλώττισης. Αποφασίζουμε λοιπόν, να κάνουμε χρήση του flag -O3, δίνοντας στο μεταγλωττιστή το μέγιστο βαθμό ελευθερίας, να πραγματοποιήσει διάφορες βελτιστοποιήσεις στον πηγαίο κώδικα.

Program Parameters	
Flag	**Description**
-p	grid size
-f	number of k-NN
-i	number of iterations
-w	object & point file paths

Πίνακας 4.2: Παράμετροι προγράμματος

Τέλος, όσον αφορά τη διεξαγωγή τόσο των δοκιμών, όσο και των τελικών πειραμάτων, εργαζόμαστε στο σύστημα csl-artemis του Τμήματος, το οποίο έχει τα εξής χαρακτηριστικά:

- Μοντέλο επεξεργαστή: Intel Xeon E5-2683 v4
- Συχνότητα επεξεργαστή: 2.10GHz
- Επάρκεια σε μνήμη RAM: 128 GB

4.1.3 Διαδικασία Πειραμάτων

Το μόνο κομμάτι του setup που απομένει να δούμε, είναι η διαδικασία που ακολουθούμε στα πειράματά μας. Για το σκοπό αυτό επιλέγουμε να αξιολογήσουμε ξεχωριστά τις επιδόσεις της ευρετηρίασης και του αλγόριθμου διεκπεραίωσης των ερωτημάτων.

Ξεκινάμε λοιπόν με την ευρετηρίαση των αντικειμένων. Στο στάδιο αυτό, για κάθε αρχείο αντικειμένων, καθώς και για κάθε μέγεθος πλεγματικής διαμέρισης, κάνουμε δώδεκα εκτελέσεις του εν λόγω κώδικα. Από τις μετρήσεις που συλλέγουμε, απορρίπτουμε αυτές που παραπέμπουν στην καλύτερη και στη χειρότερη επίδοση. Με τις εναπομείναντες δέκα μετρήσεις, υπολογίζουμε το μέσο χρόνο ευρετηρίασης, το οποίο είναι και το τελικό μας ζητούμενο.

Παρόμοια είναι η διαδικασία και στο στάδιο διεκπεραίωσης των ερωτημάτων. Στην ουσία, για κάθε ερώτημα από τα δέκα χιλιάδες που έχουμε στη διάθεσή μας, πραγματοποιούμε δώδεκα επαναλήψεις του , αφαιρούμε τον καλύτερο και το χειρότερο χρόνο, και στο τέλος

βρίσκουμε το μέσο χρόνο εκτέλεσής του. Με βάση τα αποτελέσματα που προκύπτουν, επιλέγουμε να μην σταματήσουμε σε αυτό το στάδιο την αξιολόγηση των ερωτημάτων, καθώς υπάρχουν σημαντικές αποκλίσεις στους χρόνους τους.

Αναζητώντας σε βάθος την αιτία που οδηγεί στις διαφοροποιήσεις των αποτελεσμάτων, ελέγχουμε το πλήθος των κελιών που επισκεπτόμαστε σε κάθε εκτελεσμένο ερώτημα. Από αυτή την ενέργεια, προκύπτει το συμπέρασμα ότι η πλειοψηφία του συνόλου αυτού, οδηγείται στην εξέταση δεκάδων κελιών. Ωστόσο, υπάρχει και ένα ποσοστό, στο οποίο εξετάζονται εκατοντάδες ή και χιλιάδες κελιά. Για να καταλήξουμε λοιπόν, σε ένα πλήρες πόρισμα σχετικά με την αποδοτικότητα του αλγόριθμού μας, αποφασίζουμε να υπολογίζουμε το μέσο χρόνο εκτέλεσης ενός ερωτήματος, αθροίζοντας τους τελικούς χρόνους όλων των ερωτημάτων και διαιρώντας τους με το πλήθος τους.

4.2 Αξιολόγηση διαφορετικών εκδόσεων

Όπως είδαμε στο κεφάλαιο 3, στην πορεία υλοποίησης της προσέγγισής μας, έχουμε αναπτύξει διάφορες εκδόσεις, καθεμία από τις οποίες χαρακτηρίζεται ικανοποιητική ως προς την ορθότητα των αποτελεσμάτων της. Στην ενότητα αυτή λοιπόν, προχωράμε στην αξιολόγηση των επιδόσεων τους.

Όλες οι διαφοροποιήσεις στη πρότασή μας, σχετίζονται αποκλειστικά με το στάδιο αναζήτησης γειτονικών κελιών, όπου έχουν αναπτυχθεί τέσσερις διαφορετικές μέθοδοι (υποενότητα 3.3.5). Αποφασίζουμε λοιπόν, να αξιολογήσουμε καθεμία από αυτές, εφαρμόζοντάς την στον αλγόριθμό μας, και μετρώντας το μέσο χρόνο διεκπεραίωσης ενός ερωτήματος, ώστε να λάβουμε χρήσιμες πληροφορίες σχετικά με την αποδοτικότητά της. Εδώ αξίζει να σημειωθεί ότι μελετάμε μόνο το στάδιο του αλγόριθμου, καθώς μέχρι και την ευρετηρίαση των αντικειμένων, η προσέγγισή μας παραμένει ίδια και ανεξάρτητη από τις μεθόδους που εφαρμόζουμε.

Στο άνω και κάτω μέρος του Πίνακα 4.3, παρουσιάζουμε τους μέσους χρόνους διεκπεραίωσης ενός ερωτήματος, για όλους τους δυνατούς συνδυασμούς αντικειμένων και τιμών k, που επιλέγουμε να ασχοληθούμε. Είναι ορατό, πως η μέθοδος με την εφαρμογή των ορθογώνιων στην αναζήτηση γειτονικών κελιών, επικρατεί έναντι των υπόλοιπων τριών, προσφέροντας τη καλύτερη επίδοση στην υλοποίησή μας. Από τα αποτελέσματα αυτά, πέρα από την υπεροχή της τέταρτης μεθόδου, μπορούμε να δούμε κάποια επιπλέον χρήσιμα συμπερά-

σματα σχετικά με την περιγραφή των μεθόδων, που είδαμε στο προηγούμενο κεφάλαιο.

Mean Querying Times (μs)						
Method	**T1**			**T2**		
	k = 10	**k = 30**	**k = 50**	**k = 10**	**k = 30**	**k = 50**
Method 1	2.08	3.95	5.75	4.51	7.12	9.55
Method 2	2.09	4.01	5.81	4.57	7.21	9.61
Method 3	1.49	2.73	3.87	2.83	4.51	5.98
Method 4	**1.39**	**2.56**	**3.59**	**2.64**	**4.24**	**5.57**

Mean Querying Times (μs)						
Method	**T3**			**T4**		
	k = 10	**k = 30**	**k = 50**	**k = 10**	**k = 30**	**k = 50**
Method 1	5.54	7.09	8.62	9.94	11.94	13.63
Method 2	5.48	6.98	8.68	9.95	12.01	13.58
Method 3	3.19	4.56	5.79	5.25	7.04	8.54
Method 4	**2.95**	**4.26**	**5.39**	**4.89**	**6.57**	**7.92**

Πίνακας 4.3: Μέσοι χρόνοι ερωτήματος για καθεμία μέθοδο

Ξεκινάμε λοιπόν, με τις δύο πρώτες μεθόδους, οι οποίες είναι και αυτές που στηρίζονται στη λογική της λίστας επισκέψεων. Συγκρίνοντας τις μετρήσεις τους, βλέπουμε ότι υπάρχουν μικρές αποκλίσεις σε καθένα από τα πειράματα, τις οποίες θεωρούμε αμελητέες. Αυτό σημαίνει πως η απόφαση να αυξήσουμε το πλήθος των γειτόνων, στις περιπτώσεις των κελιών του σταυρού (δεύτερη μέθοδος), δεν λειτουργεί επιβαρυντικά στην επίδοση του αλγόριθμου.

Συνεχίζουμε την ανάλυσή μας, συγκρίνοντας αυτή τη φορά, τους χρόνους της τρίτης μεθόδου με τους αντίστοιχους των προηγούμενων δύο. Σε αυτή την περίπτωση εντοπίζονται σημαντικές αποκλίσεις, οι οποίες ακολουθούν αυξητική τάση, καθώς προχωράμε σε δοκιμές τόσο με μεγαλύτερες τιμές του k, όσο και με μεγαλύτερα σύνολα αντικειμένων. Προφανώς, το κέρδος που επιτυγχάνεται με τη τρίτη μέθοδο, συνδέεται εξ ολοκλήρου με την επιλογή μας να αποφύγουμε τη χρήση λίστας επισκέψεων, αλλάζοντας στην ουσία τον τρόπο με τον οποίο αναζητάμε τα γειτονικά κελιά.

Τέλος, όσον αφορά τη τέταρτη μέθοδο, παρότι έχει και αυτή ίδια συμπεριφορά έναντι

των δύο πρώτων μεθόδων, αυτό που έχει ενδιαφέρον είναι η σχέση των χρόνων της, έναντι των αντίστοιχων της τρίτης μεθόδου. Συγκεκριμένα, από τον πίνακα μπορούμε να δούμε, ότι η εφαρμογή των ορθογώνιων, προσφέρει στον αλγόριθμο μία επιτάχυνση της τάξης του 7-7.5%, η οποία συνδέεται αποκλειστικά με την ελεγχόμενη εξέταση μία γειτονιάς κελιών, όταν και αν αυτή χρήζει αναγκαία.

4.3 Αξιολόγηση έναντι αντίπαλης προσέγγισης

4.3.1 Οργάνωση αντίπαλης προσέγγισης

Το επόμενο στάδιο μετά τον εντοπισμό της καλύτερης έκδοσής μας, είναι η αξιολόγησή της έναντι κάποιας καθιερωμένης προσέγγισης. Καθώς σε όλα τα χωρικά ερωτήματα, και ιδίως σε αυτά που αναζητούνται οι k πλησιέστεροι γείτονες, έχει επικρατήσει η εφαρμογή του R-tree ως μέθοδος ευρετηρίασης, αποφασίζουμε να χρησιμοποιήσουμε ως μέτρο σύγκρισης, κάποια υλοποίηση, βασισμένη σε αυτό τον τρόπο χωρικής πρόσβασης.

Για την ανάπτυξη λοιπόν της μελέτης αυτής, επιλέγουμε να κάνουμε χρήση της υλοποίησης του R-tree, που προσφέρεται από τις βιβλιοθήκες της Boost [29]. Η συγκεκριμένη επιλογή έγινε με τα εξής κριτήρια:

- Είναι απλή όσον αφορά το κομμάτι της εγκατάστασης των πακέτων που χρειάζονται για να εκτελεστεί.
- Πρόκειται για open source υλοποίηση, με φιλικό API που διευκολύνει την ανάπτυξη πηγαίου κώδικα.
- Εγγυάται ορθότητα ως προς το κομμάτι των αποτελεσμάτων, γεγονός που αξιοποιούμε για τον επανέλεγχο της δικής μας προσέγγισης.

Στην επίσημη ιστοσελίδα της Boost, υπάρχει αναλυτικός οδηγός [30], αλλά και ενδεικτική υλοποίηση σχετικά με την ευρετηρίαση χωρικών αντικειμένων και την ανάπτυξη ερωτημάτων. Από τη στιγμή που μάς απασχολεί η σύγκριση των επιδόσεων, αποφασίζουμε να εφαρμόσουμε διάφορες δοκιμές γύρω από τις παραμέτρους της εν λόγω υλοποίησης, προκειμένου να βρούμε το κατάλληλο setup, με τους καλύτερους δυνατούς χρόνους.

Ξεκινάμε λοιπόν τις δοκιμές μας, ασχολούμενοι εξ ολοκλήρου με τον τρόπο ευρετηρίασης των αντικειμένων. Η συγκεκριμένη βιβλιοθήκη, όσον αφορά την καταγραφή των

αντικειμένων, παρέχει στους χρήστες, τη δυνατότητα να αναπτύξουν τρεις διαφορετικούς αλγόριθμους: (i) τον Linear, (ii) τον Quadratic και (iii) τον R*-tree . Και οι τρεις έχουν ως στόχο να περιορίσουν τυχόν επικαλύψεις μεταξύ των κόμβων του δέντρου, ωστόσο ακολουθούν διαφορετική τεχνική διάσπασης, η οποία οδηγεί εν τέλει και σε διαφορετική δομή στο εσωτερικό του δέντρου.

Από τη δοκιμή και των τριών αλγορίθμων, καταλήγουμε στο συμπέρασμα ότι η εφαρμογή του R*-tree, προσφέρει την καλύτερη δυνατή επίδοση, πράγμα που επιβεβαιώνεται και από τους ίδιους τους προγραμματιστές της βιβλιοθήκης[31]. Εκτός από τους προαναφερθέντες αλγόριθμους, γίνεται αναφορά και στη χρήση μίας ακόμη τεχνικής, ονόματι Bulk Loading, στην οποία συναντάται η εφαρμογή του packing αλγόριθμου ([32, 33]). Η βασική της ιδέα στηρίζεται στη γρήγορη αποθήκευση των αντικειμένων στο εσωτερικό του δέντρου, με τέτοιο τρόπο ώστε να επιταχύνεται και η διαδικασία διεκπεραίωσης των ερωτημάτων. Επιλέγουμε λοιπόν, να συμπεριλάβουμε στην αξιολόγηση τόσο την εφαρμογή του R*-tree, όσο και αυτή του Bulk Loading, για να μελετήσουμε τη συμπεριφορά της δικής μας πρότασης, έναντι και των δυο αυτών προσεγγίσεων.

Μία ακόμη παράμετρος, η οποία συνδέεται με την ευρετηρίαση των αντικειμένων, είναι ο μέγιστος αριθμός στοιχείων ανά κόμβο του δέντρου. Είναι σαφές, πως για διαφορετικό βαθμό χωρητικότητας ανά κόμβο, καταλήγουμε και σε διαφορετική εσωτερική δομή του δέντρου. Αυτό σε μετέπειτα στάδιο, επηρεάζει και την αποδοτικότητα στον εντοπισμό των αντικειμένων, από τη στιγμή που διαφοροποιούνται οι διαπεράσεις που χρειάζονται για να ανακτηθεί το επιθυμητό υποσύνολο τους. Δοκιμάζουμε λοιπόν, διαφορετικές τιμές της παραμέτρου στο εύρος [1-100], και βρίσκουμε ότι για μέγιστο αριθμό στοιχείων ίσο με 16, το R-tree παρουσιάζει στο σύνολο (ευρετηρίαση και διεκπεραίωση ερωτημάτων), την καλύτερη επίδοσή του.

4.3.2 Αξιολόγηση χρόνων

Έχοντας ολοκληρώσει την περιγραφή του setup της αντίπαλης προσέγγισης, φτάνουμε στο στάδιο της αξιολόγησης των χρόνων. Αρχίζουμε λοιπόν, με το κομμάτι της ευρετηρίασης, υπολογίζοντας τους μέσους χρόνους για κάθε σύνολο αντικειμένων που έχουμε στη διάθεσή μας. Οι μετρήσεις που προκύπτουν συνοψίζονται στον Πίνακα 4.4. Όσον αφορά τη δική μας προσέγγιση, παραθέτουμε επιπλέον τον Πίνακα 4.5, στον οποίο συμπεριλαμβάνονται τα εύρη τιμών που δοκιμάζονται ως μεγέθη της πλεγματικής διαμέρισης, τα βήματα

που ακολουθούμε για να προχωρήσουμε στην επόμενη δοκιμή, καθώς και τα μεγέθη από τα οποία προκύπτουν οι τελικοί μας χρόνοι.

Mean Indexing Times (s)				
Approaches	**T1**	**T2**	**T3**	**T4**
R*-tree	0.0252	5.9012	13.4365	42.9443
Bulk Loading	0.0024	0.6112	1.4514	5.9762
Our Approach	**0.0012**	**0.1582**	**0.4691**	**2.1254**

Πίνακας 4.4: Μέσοι χρόνοι ευρετηρίασης για κάθε προσέγγιση

Grid Sizes				
Datasests	**T1**	**T2**	**T3**	**T4**
Ranges	50 - 250	500 - 2000	500 - 2000	800 - 2400
Steps	10	100	100	100
Best Sizes	150	1000	1400	2000

Πίνακας 4.5: Δοκιμές μεγεθών πλεγματικής διαμέρισης

Με βάση τα αποτελέσματα που προκύπτουν, είναι εμφανής η υπεροχή της δικής μας μεθόδου έναντι και των δύο παραλλαγών ευρετηρίασης του R-tree. Πιο αναλυτικά, συγκρίνοντας τους χρόνους μας έναντι των αντίστοιχων της μεθόδου R*-tree, φαίνεται πως είμαστε σε θέση να πετύχουμε τουλάχιστον 20 φορές πιο γρήγορη καταγραφή των αντικειμένων, σε καθένα από τα σύνολα που εξετάζουμε.

Όσον αφορά τη σύγκριση των μετρήσεών μας σε σχέση με των αντίστοιχων της μεθόδου Bulk Loading, φαίνεται ότι υπερέχουμε και σε αυτή την περίπτωση, ωστόσο σε μικρότερη κλίμακα (τουλάχιστον 2 φορές πιο γρήγορη καταγραφή). Παρόλα αυτά, αν αναλογιστούμε ότι η εν λόγω τεχνική είναι βέλτιστη απόπειρα της αντίπαλης προσέγγισης, για ταχύτερη ευρετηρίαση, είναι ξεκάθαρο πως τα αποτελέσματα μας είναι ιδιαίτερα ικανοποιητικά.

Συνεχίζουμε την μελέτη, προχωρώντας με τη σύγκριση των μέσων χρόνων διεκπεραίωσης ενός ερωτήματος. Όπως στην προηγούμενη ενότητα, έτσι και σε αυτή, αξιολογούμε τους χρόνους για κάθε δυνατό συνδυασμό συνόλων αντικειμένων και τιμών k, που έχουμε επιλέξει να χρησιμοποιήσουμε. Τα τελικά αποτελέσματα συγκεντρώνονται στο άνω και κάτω μέρος του Πίνακα 4.6.

Mean Querying Times (μs)						
Approaches	**T1**			**T2**		
	k = 10	**k = 30**	**k = 50**	**k = 10**	**k = 30**	**k = 50**
R*-tree	1.87	3.58	5.27	3.69	7.13	10.97
Bulk Loading	2.27	4.15	6.05	3.75	6.45	9.11
Our Approach	**1.39**	**2.56**	**3.59**	**2.64**	**4.24**	**5.57**

Mean Querying Times (μs)						
Approaches	**T3**			**T4**		
	k = 10	**k = 30**	**k = 50**	**k = 10**	**k = 30**	**k = 50**
R*-tree	3.69	7.18	11.22	5.72	11.85	18.44
Bulk Loading	3.75	5.89	8.58	5.18	9.18	13.51
Our Approach	**2.95**	**4.26**	**5.39**	**4.89**	**6.57**	**7.92**

Πίνακας 4.6: Μέσοι χρόνοι ερωτήματος για κάθε προσέγγιση

Όπως στο στάδιο της ευρετηρίασης, έτσι και στη διεκπεραίωση των ερωτημάτων, είναι εμφανές ότι η αλγοριθμική διαδικασία, που αναπτύσσουμε για την αναζήτηση πλησιέστερων γειτόνων, επικρατεί σημαντικά έναντι της αντίστοιχης που εφαρμόζεται στις R-tree προσεγγίσεις. Μάλιστα, καθώς προχωράμε σε μεγαλύτερες τιμές του k, είμαστε σε θέση να διεκπεραιώσουμε ερωτήματα, επιταχύνοντας ακόμα περισσότερο τη διαδικασία. Το αποτέλεσμα αυτό είναι λογικό, αν αναλογιστούμε ότι στη λογική των R-tree προσεγγίσεων χρειάζεται να πραγματοποιηθούν αρκετές διαπεράσεις στους κόμβους του δέντρου. Προφανώς τέτοιου είδους διαπεράσεις κοστίζουν πολύ περισσότερο, σε σχέση με αυτές που πραγματοποιούμε εμείς, οι οποίες αντιστοιχούν σε συνεχόμενες θέσεις ενός πίνακα.

Είναι γεγονός ότι οι χρόνοι των ερωτημάτων, μάς απασχολούν περισσότερο από αυτούς της ευρετηρίασης. Αυτό συμβαίνει καθώς σε δεδομένα στατικού χαρακτήρα (όπως και στην περίπτωσή μας), η διαδικασία της ευρετηρίασης καλείται να πραγματοποιηθεί μία φορά. Συνεπώς, ακόμα και αν ο χρόνος της κυμαίνεται κοντά σε αυτούς του R-tree, εξακολουθεί να θεωρείται ικανοποιητική. Αυτό που πραγματικά έχει σημασία να αξιολογηθεί είναι ο ρυθμός με τον οποίο φέρνουμε εις πέρας τα ερωτήματα που υποβάλλονται. Βάσει λοιπόν αυτού, επιλέγουμε να παρουσιάσουμε στο άνω και κάτω μέρος του Πίνακα 4.7, το πλήθος των ερωτημάτων που διεκπεραιώνονται ανά δευτερόλεπτο. Από τα αποτελέσματά που προκύπτουν,

γίνεται αντιληπτό πως εκτελούμε ερωτήματα με αρκετά μεγαλύτερη συχνότητα, γεγονός που επιβεβαιώνει για ακόμη μία φορά την επικράτησή μας έναντι του R-tree.

Number of queries / second						
Approaches	**T1**			**T2**		
	k = 10	**k = 30**	**k = 50**	**k = 10**	**k = 30**	**k = 50**
R*-tree	534.8K	279.3K	189.8K	271K	140.3K	91.2K
Bulk Loading	440.5K	240.9K	165.3K	266.7K	155K	109.8K
Our Approach	**719.4K**	**390.6K**	**278.6K**	**378.8K**	**235.8K**	**179.5K**

Number of queries / second						
Approaches	**T3**			**T4**		
	k = 10	**k = 30**	**k = 50**	**k = 10**	**k = 30**	**k = 50**
R*-tree	271K	139.3K	89.1K	174.8K	84.4K	54.2K
Bulk Loading	266.7K	169.8K	116.6K	193.1K	108.9K	74.1K
Our Approach	**339K**	**234.7K**	**185.5K**	**204.5K**	**152.2K**	**126.3K**

Πίνακας 4.7: Αριθμός εκτελεσμένων ερωτημάτων ανά δευτερόλεπτο

Κεφάλαιο 5

Συμπεράσματα

5.1 Σύνοψη και συμπεράσματα

Έχοντας δει την ανάλυση των προηγούμενων κεφαλαίων, είναι πλέον κατανοητό πως η προσέγγισή μας είναι ικανή να διαχειριστεί την ευρετηρίαση μεγάλης κλίμακας συνόλων αντικειμένων, στο επίπεδο της Κύριας Μνήμης. Μέσω του αλγόριθμου που αναπτύσσουμε, σε συνδυασμό με τη διεπίπεδη διαμέριση του χώρου, είμαστε σε θέση να πραγματοποιήσουμε την ορθή διεκπεραίωση ερωτημάτων k πλησιέστερων γειτόνων. Πέρα όμως από την ορθότητα, οι μέθοδοι που εισάγουμε για την αποφυγή διπλότυπων εμφανίσεων, τόσο των αντικειμένων, όσο και των κελιών, εξασφαλίζουν και το ζητούμενο της αποδοτικότητας στην εκτέλεση των ερωτημάτων. Συγκεκριμένα, τα αποτελέσματα της πειραματικής αξιολόγησης, επιβεβαιώνουν στο σύνολό τους, την επικράτηση της προσέγγισής μας, τόσο στο στάδιο της καταγραφής όσο και στο στάδιο της διεκπεραίωσης, από τη στιγμή που πετυχαίνουμε καλύτερους μέσους χρόνους, έναντι και των δύο παραλλαγών του R-tree, με τις οποίες συγκρινόμαστε.

5.2 Μελλοντικές επεκτάσεις

Μέχρι στιγμής, η προσέγγισή μας βασίζεται στην ενασχόληση με MBRs αντικειμένων, πραγματοποιώντας την εκτέλεση ερωτημάτων πάνω σε αυτά (filter step). Έτσι λοιπόν, μία περίπτωση επέκτασης της υλοποίησής μας, είναι η ανάπτυξη και του refinement step, με το οποίο θα εξετάζεται κατά πόσο η επιλογή των αποτελεσμάτων με βάση τα MBRs, επιβεβαιώνεται και για τις πραγματικές γεωμετρίες τους.

Όσον αφορά την υπάρχουσα μορφή της υλοποίησης, ιδιαίτερο ενδιαφέρον έχει η απόπειρα παραλληλοποίησής της, με σκοπό να μελετηθεί η επίδοσή της, σε πολυνηματικές εκτελέσεις. Από τη μία, στο κομμάτι της ευρετηρίασης, όπου στηριζόμαστε στη λογική των SOP προσεγγίσεων, είδαμε (κεφάλαιο 2) ότι η ανεξαρτησία των διαμερισμένων τμημάτων τους, ευνοεί την προσπάθεια παραλληλοποίησής τους. Από την άλλη, στο στάδιο εκτέλεσης των ερωτημάτων, πραγματοποιούμε αρκετές πράξεις υπολογισμού αποστάσεων, δημιουργώντας την ανάγκη ανάθεσης αυτού του σταδίου σε πολλαπλά νήματα, ώστε να περιοριστεί σε ένα βαθμό το επιβαρυντικό κόστος.

Βιβλιογραφία

[1] Vector and raster data models. https://www.studypool.com/documents/4463299/vector-and-raster-data-models.

[2] Spatial relationships and filtering. https://docs.oracle.com/database/121/SPATL/spatial-relationships-and-filtering.htm#SPATL460.

[3] N. Mamoulis. *Spatial Data Management*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[4] Shashi Shekhar and Sanjay Chawla. *Spatial databases - a tour.* 01 2003.

[5] Zhexue Huang. Topological spatial relations and operators. In *Proc. of the 17th ISPRS Congress, Washington DC*, 1992.

[6] David J. Maguire Paul A. Longley, Mike Goodchild and David W. Rhind. *Geographic Information Systems and Science*. Wiley Publishing, 3rd edition, 2010.

[7] Spiros Skiadopoulos. *Directional Relations*, pages 1–7. Springer International Publishing, Cham, 2015.

[8] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction*. Monographs in Computer Science. Springer New York, 2012.

[9] Hanan Samet. *The design and analysis of spatial data structures*, volume 85. Addison-wesley Reading, MA, 1990.

[10] Hanan Samet. *Foundations of multidimensional and metric data structures.* Morgan Kaufmann series in data management systems. Academic Press, 2006.

[11] Suprio Ray, Rolando Blanco, and Anil K. Goel. Supporting location-based services in a main-memory database. In *2014 IEEE 15th International Conference on Mobile Data Management*, volume 1, pages 3–12, 2014.

[12] Darius Sidlauskas, Simonas Šaltenis, Christian Christiansen, Jan Johansen, and Donatas Saulys. Trees or grids? indexing moving objects in main memory. pages 236–245, 11 2009.

[13] Matthaios Olma, Farhan Tauheed, Thomas Heinis, and Anastasia Ailamaki. Block: Efficient execution of spatial range queries in main-memory. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, pages 1–12, 2017.

[14] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, sep 1975.

[15] Raphael A Finkel and Jon Louis Bentley. Quad trees a data structure for retrieval on composite keys. *Acta informatica*, 4(1):1–9, 1974.

[16] Ahmed Eldawy and Mohamed F Mokbel. Spatialhadoop: A mapreduce framework for spatial data. In *2015 IEEE 31st international conference on Data Engineering*, pages 1352–1363. IEEE, 2015.

[17] Dong Xie, Feifei Li, Bin Yao, Gefei Li, Liang Zhou, and Minyi Guo. Simba: Efficient in-memory spatial analytics. In *Proceedings of the 2016 international conference on management of data*, pages 1071–1085, 2016.

[18] Jia Yu, Zongsi Zhang, and Mohamed Sarwat. Spatial data management in apache spark: The geospark perspective and beyond. *Geoinformatica*, 23(1):37–78, jan 2019.

[19] J.-P. Dittrich and B. Seeger. Data redundancy and duplicate detection in spatial join processing. In *Proceedings of 16th International Conference on Data Engineering (Cat. No.00CB37073)*, pages 535–546, 2000.

[20] Walid G. Aref and Hanan Samet. Hashing by proximity to process duplicates in spatial databases. In *Proceedings of the Third International Conference on Information and Knowledge Management*, CIKM '94, page 347–354, New York, NY, USA, 1994. Association for Computing Machinery.

[21] D. Tsitsigkos, K. Lampropoulos, P. Bouros, N. Mamoulis, and M. Terrovitis. A two-layer partitioning for non-point spatial data. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 1787–1798, Los Alamitos, CA, USA, apr 2021. IEEE Computer Society.

[22] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. *SIGMOD Rec.*, 14(2):47–57, jun 1984.

[23] Donghui Zhang, Kenneth Paul Baclawski, and Vassilis J. Tsotras. *B+-Tree*, pages 197–200. Springer US, Boston, MA, 2009.

[24] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The r*-tree: an efficient and robust access method for points and rectangles. In *Proceedings of the 1990 ACM SIGMOD international conference on Management of data - SIGMOD '90*. ACM Press, 1990.

[25] Kihong Kim, Sang K. Cha, and Keunjoo Kwon. Optimizing multidimensional index trees for main memory access. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, SIGMOD '01, page 139–150, New York, NY, USA, 2001. Association for Computing Machinery.

[26] Kyriakos Mouratidis, Dimitris Papadias, and Marios Hadjieleftheriou. Conceptual partitioning: An efficient method for continuous nearest neighbor monitoring. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 634–645, 2005.

[27] Us census bureau. `https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.2015.html`.

[28] Spatialhadoop. `http://spatialhadoop.cs.umn.edu/datasets.html`.

[29] Boost c++ libraries. `https://www.boost.org`.

[30] Boost r-tree documentation. `https://www.boost.org/doc/libs/1_59_0/libs/geometry/doc/html/geometry/reference/spatial_indexes/boost__geometry__index__rtree.html`.

[31] Boost r-tree indexing methods. `https://www.boost.org/doc/libs/1_59_0/libs/geometry/doc/html/geometry/spatial_indexes/introduction.html`.

[32] S.T. Leutenegger, M.A. Lopez, and J. Edgington. Str: a simple and efficient algorithm for r-tree packing. In *Proceedings 13th International Conference on Data Engineering*, pages 497–506, 1997.

[33] Yván J. García, Mario A. López, and Scott T. Leutenegger. A greedy algorithm for bulk loading r-trees. In *GIS '98*, 1998.

Παράρτημα

Σύνδεσμος Προτεινόμενης Υλοποίησης

Στον σύνδεσμο `https://github.com/achmichalop/Indexing-on-k-Nearest-Neighbors-Queries-in-Main-Memory` είναι ανεβασμένη στο GitHub, όλη η προτεινόμενη υλοποίηση. Επιπλέον, συμπεριλαμβάνεται οδηγός σχετικά με την οργάνωση του κώδικα, καθώς και με τον τρόπο εκτέλεσης του προγράμματος. Τέλος, υπάρχουν διαθέσιμα αρχεία για την ανάπτυξη δοκιμών.