

Τήρηση κυρτού περιβλήματος σε γεωμετρικά ρεύματα σημειακών δεδομένων

Πέτρος Βενέτης
petervenetis@gmail.com

Χρήστος Στεργίου
ch.stergiou@gmail.com

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Εθνικό Μετσόβιο Πολυτεχνείο

Περίληψη

Ο σκοπός αυτής της εργασίας είναι η μελέτη ενός αλγόριθμου τήρησης ενός προσεγγιστικού κυρτού περιβλήματος σε γεωμετρικά ρεύματα σημειακών δεδομένων. Έτσι, αφού αρχικά μελετήσουμε το πρόβλημα και αναφερθούμε στην εύρεση του κυρτού περιβλήματος ενός στατικού συνόλου σημειακών δεδομένων, εξετάζουμε κατόπιν την τήρηση ενός προσεγγιστικού κυρτού περιβλήματος ενός γεωμετρικού ρεύματος σημειακών δεδομένων. Οι πειραματικές μας μετρήσεις δείχνουν ότι ο προσεγγιστικός αλγόριθμος καταφέρνει να δώσει μια αρκετά πιστή προσέγγιση του ακριβούς περιβλήματος.

1 Εισαγωγή

Για πολλές τεχνολογικές εφαρμογές είναι πλέον απαραίτητη η ύπαρξη ενός αλγορίθμου υπολογισμού του κυρτού περιβλήματος (convex hull) ενός γεωμετρικού ρεύματος δεδομένων, το οποίο είναι το μικρότερο κυρτό πολύγωνο που περιλαμβάνει όλα τα σημεία που έχουμε εξετάσει. Ας δούμε ένα σύντομο παράδειγμα που καταδεικνύει τη σημασία που έχει ο αλγόριθμος αυτός. Ας υποθέσουμε ότι σε κάποια περιοχή έχουμε ένα εργοστάσιο το οποίο παράγει επιβλαβή προς το περιβάλλον προϊόντα. Για να αντιμετωπίσουμε τις αρνητικές συνέπειες τυχόν ρύπανσης, διαθέτουμε ορισμένους κινούμενους αισθητήρες (sensors) στην περιοχή. Στην περίπτωση που κάποιος αισθητήρας ανιχνεύσει κάποιο από τα επικίνδυνα χημικά στέλνει κατάλληλο σήμα το

οποίο περιέχει πληροφορία σχετική με το χημικό που εντόπισε καθώς και την ακριβή του θέση εκείνη τη στιγμή. Με αυτόν τον τρόπο, μπορούμε να γνωρίζουμε ανά χρονικά διαστήματα πως εξελίσσεται η διαρροή. Αν οι αισθητήρες είναι πολλοί και στέλνουν αρκετά συχνά πληροφορίες σχετικές με την εξάπλωση του χημικού, τότε σε σύντομο χρονικό διάστημα θα έχουμε πολλά δεδομένα, τα οποία συγκροτούν ένα γεωμετρικό ρεύμα δεδομένων. Το σημαντικό στο παράδειγμά μας είναι ότι θέλουμε να ξέρουμε την περιοχή στην οποία έχει πραγματοποιηθεί η εξάπλωση του χημικού, δηλαδή μέσα σε ποια όρια έχει εξαπλωθεί το χημικό. Για αυτό το λόγο, θέλουμε ανά πάσα στιγμή να γνωρίζουμε το κυρτό περίβλημα των αισθητήρων που έχουν παρατηρήσει τη διαρροή, ώστε να γνωρίζουμε το μέγεθος της εξάπλωσης του χημικού.

Το παράδειγμα που προηγήθηκε καταδεικνύει για ποιο λόγο είναι τόσο σημαντικός ο αλγόριθμος που υπολογίζει το κυρτό περίβλημα ενός ρεύματος σημειακών δεδομένων. Αλλά δεν είναι το μόνο. Ο αλγόριθμος αυτός είναι χρήσιμος και σε πολλές άλλες εφαρμογές, όπως είναι παρατήρηση της περιοχής διαβίωσης άγριων θηλαστικών ή πτηνών, παρακολούθηση μίας βιομηχανικής εγκατάστασης και σε πολλά άλλα αντίστοιχα παραδείγματα.

Το πρόβλημα υπολογισμού του κυρτού περιβλήματος για στατικά δεδομένα είναι ένα γεωμετρικό πρόβλημα, το οποίο είναι αρκετά περίπλοκο. Έχουν δοθεί κατά καιρούς διάφοροι αλγόριθμοι υπολογισμού του ακριβούς κυρτού περιβλήματος, τόσο ακριβείς όσο και προσεγγιστικοί. Ακόμα πιο δύσκολο και πολύπλοκο όμως

είναι το πρόβλημα της τήρησης του κυρτού περιβλήματος για ένα ρεύμα σημειακών δεδομένων. Για να δοθεί μία λύση απαιτείται πολύ καλή κατανόηση της γεωμετρίας του προβλήματος και συνήθως χρησιμοποιούνται προσεγγιστικοί αλγόριθμοι, καθώς η χρονική πολυπλοκότητα είναι αρκετά μεγάλη.

Το άρθρο αυτό αποτελείται από έξι (6) ενότητες. Στην ενότητα 2 περιγράφονται τα γεωμετρικά ρεύματα δεδομένων. Στην ενότητα 3 περιγράφεται το πρόβλημα του κυρτού περιβλήματος καθώς και ένας ακριβής αλγόριθμος υπολογισμού του για την περίπτωση στατικών δεδομένων, ενώ στην ενότητα 4 αναλύεται ένας ευριστικός αλγόριθμος που χρησιμοποιήσαμε για την προσεγγιστική επίλυση του προβλήματος για την περίπτωση ρεύματος σημειακών δεδομένων. Στην ενότητα 5 αναφερόμαστε στα πειράματα που εκτελέσαμε, με τα συμπεράσματά μας να αναλύονται στην ενότητα 6.

2 Γεωμετρικά ρεύματα δεδομένων

Είναι πολύ σημαντικό να αντιληφθούμε τι ακριβώς είναι ένα ρεύμα δεδομένων ώστε να κατανοήσουμε και το λόγο για τον οποίο ο χειρισμός του είναι κατά πολύ δυσκολότερος από το χειρισμό απλών ‘στατικών’ δεδομένων.

Ορισμός 1. Ρεύμα δεδομένων (data stream) είναι μία ακολουθία πλειάδων (tuples) $p_1, p_2, \dots, p_N, \dots$, όπου p_1 είναι το πρώτο στοιχείο που εισήλθε στο σύστημα και p_N το τελευταίο που έχει καταγραφεί ως τώρα.

Όπως παρατηρούμε από τον ορισμό που προηγήθηκε, στο μοντέλο που χρησιμοποιούμε δεν έχουμε καμία υπόθεση σχετικά με τη χρονική διάταξη των στοιχείων, αν και συνήθως το κάθε στοιχείο συνοδεύεται και από μία ένδειξη μιας χρονικής στιγμής που αντιστοιχεί. Δηλαδή, σε ένα πρακτικό σύστημα, εκτός από τις απαραίτητες πληροφορίες (και το πιθανό ID του στοιχείου), θα υπήρχε επιπλέον και ένα χρονόσημο (timestamp), το οποίο θα έδειχνε το χρόνο κατά τον οποίο εστάλη το συγκεκριμένο δεδομένο.

Ας ξεφύγουμε όμως τώρα από τα συνηθισμένα ρεύματα δεδομένων και ας έρθουμε στα γεωμετρικά ρεύματα δεδομένων (ή ρεύματα σημειακών δεδομένων) που

μας απασχολούν σε αυτήν την εργασία. Όπως είναι προφανές και από το όνομά τους, τα γεωμετρικά ρεύματα δεδομένων είναι και αυτά ρεύματα δεδομένων και έχουν παραπλήσιες ιδιότητες με τα ρεύματα δεδομένων. Αυτό που τα διαφοροποιεί όμως από τα συνηθισμένα ρεύματα δεδομένων είναι το γεγονός ότι η κάθε πλειάδα p_i του ρεύματος έχει και το ζευγάρι (x_i, y_i) που προσδιορίζει και τη θέση της πηγής κατά τη χρονική στιγμή i στο επίπεδο.

Τα δεδομένα από ένα ρεύμα δεδομένων καταφθάνουν με απρόβλεπτο ρυθμό, ενώ και το συνολικό τους μέγεθος είναι άγνωστο. Επίσης, το αποτέλεσμα της απαιτούμενης επεξεργασίας πρέπει πολλές φορές να παρέχεται σε πραγματικό χρόνο και μάλιστα για ερωτήματα διαρκείας (continuous queries), τα οποία παραμένουν ενεργά για μεγάλα χρονικά διαστήματα. Επιπλέον, για λόγους ταχύτητας πρέπει ο υπολογισμός να γίνεται στη μνήμη, η οποία όμως σίγουρα θεωρείται ότι είναι πεπερασμένη και μάλλον ανεπαρκής για τόσο μεγάλο όγκο δεδομένων. Για αυτούς τους λόγους είναι απαραίτητο να χρησιμοποιηθούν αλγόριθμοι ενός περάσματος (one-pass algorithms), οι οποίοι εξετάζουν τα στοιχεία με τη σειρά που αυτά έρχονται στην κύρια μνήμη και έπειτα—χωρίς να τα αποθηκεύσουν κάπου—συνεχίζουν με το επόμενο στοιχείο.

Λόγω του πολύ μεγάλου αριθμού στοιχείων που το σύστημα είναι πιθανό να δεχτεί και της δυσκολίας που έχουμε να τα αποθηκεύσουμε στο σύνολό τους, συνήθως διατηρούμε προσεγγιστικές συνοψείς (synopses) του ρεύματος χωρικών δεδομένων. Η σύνοψη έχει ως σκοπό τον υπολογισμό χρήσιμων προτύπων που ενυπάρχουν στα δεδομένα και τη γενίκευση των χαρακτηριστικών του. Το σκοπό αυτό έχει και ο υπολογισμός του κυρτού περιβλήματος.

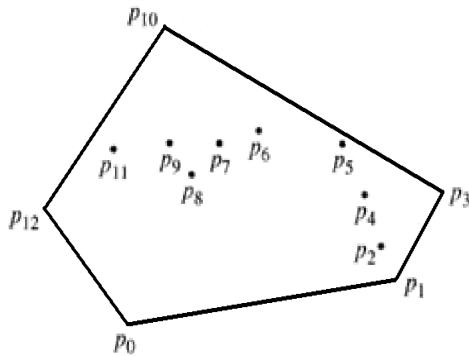
Το κυρτό περίβλημα είναι ιδανικό για την αναπαράσταση του εξωτερικού περιβλήματος των σημειακών καταγραφών. Για αυτό το λόγο και μπορεί να δώσει πολύ αδρά το σχήμα και την χωρική έκταση των δεδομένων. Ο ακριβής υπολογισμός όμως του κυρτού περιβλήματος κοστίζει αρκετά όπως θα δούμε παρακάτω και για αυτό το λόγο προτιμούμε να υπολογίζουμε ένα προσεγγιστικό κυρτό περίβλημα—με ελεγχόμενο σφάλμα εν σχέσει με το ακριβές—το οποίο όμως υπολογίζεται σαφώς ταχύτερα.

3 Κυρτό περίβλημα στατικών σημειακών θέσεων

3.1 Γενικά για το κυρτό περίβλημα

Το κυρτό περίβλημα ενός συνόλου (στατικών) σημείων P , είναι η κοινή τομή των κυρτών πολυγώνων που καθένα περικλείει όλα τα σημεία του P . Φυσικά υπάρχουν και πολλοί άλλοι ισοδύναμοι ορισμοί, όπως ο ακόλουθος, ο οποίος είναι και ο πιο διαδεδομένος.

Ορισμός 2. Το κυρτό περίβλημα ενός συνόλου σημείων P ορίζεται ως το ελάχιστο κυρτό σύνολο που περικλείει όλα τα σημεία του συνόλου P .



Σχήμα 1: Το κυρτό περίβλημα ενός συνόλου σημείων

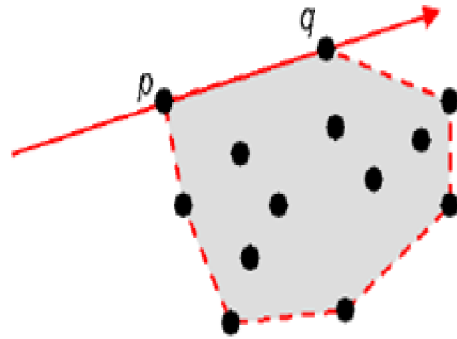
Σύμφωνα και με τους δύο ορισμούς που μόλις δόθηκαν, αλλά και με όλους τους ισοδύναμους, στο Σχήμα 1 βλέπουμε ότι το πολύγωνο με κορυφές $p_1, p_3, p_{10}, p_{12}, p_0$ είναι το κυρτό περίβλημα του συνόλου των σημείων p_0, p_1, \dots, p_{12} .

Γιατί όμως είναι τόσο σημαντικός ο υπολογισμός του κυρτού περιβλήματος ενός συνόλου σημείων; Η εμπειρία έχει δείξει ότι είναι πολύ χρήσιμο σε διάφορες εφαρμογές να είμαστε σε θέση να ανιχνεύουμε σε διάφορα ρεύματα δεδομένων ορισμένες ακραίες τιμές (outliers) που πιθανόν να έχουν, για αυτό και το κυρτό περίβλημα είναι ιδανικό για την αναπαράσταση του εξωτερικού περιβλήματος των διαφόρων σημειακών καταγραφών. Δίνει, σε αδρή μορφή, τόσο το γεωμετρικό σχήμα που έχουν τα δεδομένα όσο και τη χωρική έκταση που καταλαμβάνουν τα στοιχεία. Όταν γνωρίζουμε το κυρτό

περίβλημα ενός συνόλου σημείων, μπορούμε εύκολα να δώσουμε απάντηση στα ακόλουθα ερωτήματα ή στους ακόλουθους υπολογισμούς:

- Υπολογισμό της διαμέτρου του συνόλου των σημείων. Ως διάμετρος ενός συνόλου σημείων ορίζεται η μέγιστη απόσταση μεταξύ οποιωνδήποτε δύο σημείων του συνόλου.
- Υπολογισμό του πλάτους του συνόλου των σημείων. Ως πλάτος ορίζεται η ελάχιστη απόσταση μεταξύ δύο παράλληλων ευθειών οι οποίες περικλείουν όλα τα σημεία του συνόλου.
- Υπάρχει γραμμικός διαχωρισμός μεταξύ δύο συνόλων σημείων. Δηλαδή, υπάρχει κάποια ευθεία η οποία μπορεί να διαχωρίσει πλήρως δύο σύνολα από σημεία;
- Βρίσκεται κάποιο σημείο εντός της περιοχής κάποιου δεδομένου συνόλου στατικών σημείων;

3.2 Αλγόριθμοι υπολογισμού του ακριβούς κυρτού περιβλήματος



Σχήμα 2: Όλα τα σημεία του συνόλου μας βρίσκονται δεξιά από το διάνυσμα \vec{pq}

Λόγω της ιδιαίτερης χρησιμότητας του κυρτού περιβλήματος ενός συνόλου σημείων έχουν αναπτυχθεί κατά καιρούς πολλοί αλγόριθμοι για τον υπολογισμό του. Κατ' αρχάς, πρέπει να αναφέρουμε ότι έχει αποδειχθεί ότι η πολυπλοκότητα του προβλήματος του υ-

πολογισμού του κυρτού περιβλήματος εκφρασμένη συναρτήσει του αριθμού n των σημειακών θέσεων είναι $\Omega(n \log n)$, επομένως δεν είναι δυνατόν να βρεθεί αλγόριθμος, ο οποίος να υπολογίζει το ακριβές κυρτό περίβλημα και να έχει πολυπλοκότητα μικρότερη από $O(n \log n)$.

Ένας από τους πρώτους αλγόριθμους που δόθηκαν για τον ζητούμενο υπολογισμό βασίζεται στην εξής απλή παρατήρηση: εάν διατρέχουμε τα σημεία του κυρτού πολυγώνου ωρολογιακά, τότε για κάθε ακμή του πολυγώνου \overline{pq} ισχύει ότι όλες οι σημειακές θέσεις βρίσκονται στα δεξιά του διανύσματος αυτού. Αντιστρόφως, αν όλα τα σημεία του συνόλου μας βρίσκονται στα δεξιά του διανύσματος \overline{pq} , τότε το \overline{pq} είναι πλευρά του κυρτού περιβλήματος (όπου τα p και q είναι σημεία του συνόλου μας), όπως εποπτικά φαίνεται στο Σχήμα 2.

Έτσι έχουμε τον ακόλουθο απλό αλγόριθμο που λύνει το πρόβλημα σε χρόνο $O(n^3)$: Για κάθε ζεύγος (p, q) σημείων του συνόλου P ελέγχουμε αν όλα τα υπόλοιπα σημεία του P βρίσκονται στα δεξιά του διανύσματος \overline{pq} . Αν βρίσκονται, τότε το \overline{pq} αποτελεί μία πλευρά του κυρτού περιβλήματος που αναζητούμε.

Έχουμε $n \times (n-1)$ ζεύγη να εξετάσουμε και για κάθε ζεύγος πρέπει να γίνει έλεγχος για τα υπόλοιπα $n-2$ σημεία. Τέλος, τα σημεία πρέπει να ταξινομηθούν για να έχουμε τις πλευρές του κυρτού περιβλήματος ωρολογιακά (ή ανωρολογιακά, χωρίς αυτό να έχει κάποια σημασία). Το πρώτο σκέλος έχει κόστος $O(n^3)$ και το δεύτερο $O(n \log n)$, οπότε το συνολικό κόστος του αλγορίθμου είναι $O(n^3)$.

Ο αλγόριθμος αυτός βέβαια δεν θα μπορούσε να χρησιμοποιηθεί σε πρακτικές εφαρμογές με μεγάλο αριθμό χωρικών δεδομένων, λόγω του υψηλού του υπολογιστικού κόστους. Για αυτό το λόγο αναπτύχθηκαν άλλοι αλγόριθμοι, οι οποίοι εκτελούν τον υπολογισμό αρκετά ταχύτερα. Ένας από αυτούς τους αλγόριθμους είναι ο αλγόριθμος του Andrews που έχει πολυπλοκότητα $O(n \log n)$, στον οποίο και θα αναφερθούμε στη συνέχεια.

3.3 Ο αλγόριθμος του Andrews

Ο αλγόριθμος του Andrews τρέχει σε χρόνο $O(n \log n)$, κατασκευάζοντας ένα μέρος του κυρτού περιβλήματος και αναθεωρώντας το κατά βήματα, ώσπου να κατασκευάσει ολόκληρο το περίβλημα.

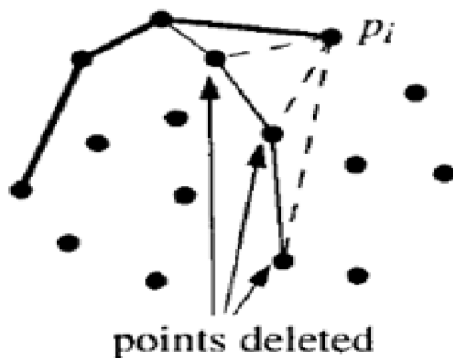
Πιο συγκεκριμένα, η μέθοδος αυτή κατασκευάζει το κυρτό περίβλημα σε δύο περάσματα: το πρώτο για το άνω μέρος του κυρτού περιβλήματος και το δεύτερο για το κάτω μέρος του. Ο αλγόριθμος βασίστηκε στην ιδέα ότι αν διατρέξουμε όλες τις ακμές του κυρτού πολυγώνου που κατασκευάζεται ωρολογιακά, τότε έχουμε μόνο δεξιές στροφές, δηλαδή μία πλευρά στρέφεται πάντα προς τα δεξιά ως προς την προηγούμενη της.

Αλγόριθμος AndrewConvexHull(P)

Input: Ένα σύνολο P από σημεία στο επίπεδο.

Output: Μία λίστα από τα σημεία του κυρτού περιβλήματος του συνόλου P σε ωρολογιακή σειρά.

1. Ταξινόμησε τα σημεία σύμφωνα με την τετμημένη τους και πάρε την ακολουθία p_1, p_2, \dots, p_n .
2. Βάλε τα σημεία p_1 και p_2 στη λίστα L_{upper} .
3. Για $i \leftarrow 3$ μέχρι n
 - (α') Βάλε το p_i στη λίστα L_{upper} .
 - (β') Όσο η λίστα L_{upper} περιέχει περισσότερα από δύο σημεία και τα τελευταία τρία σημεία που περιέχει δεν κάνουν δεξιά στροφή
 - i. Βγάλε το μεσαίο στοιχείο των τριών τελευταίων στοιχείων της λίστας L_{upper} .
4. Βάλε τα σημεία p_n και p_{n-1} στη λίστα L_{lower} με το p_n να είναι το πρώτο στοιχείο της λίστας.
5. Για $i \leftarrow n-2$ μέχρι 1
 - (α') Βάλε το p_i στη λίστα L_{lower} .
 - (β') Όσο η λίστα L_{lower} περιέχει περισσότερα από δύο σημεία και τα τελευταία τρία σημεία που περιέχει δεν κάνουν δεξιά στροφή
 - i. Βγάλε το μεσαίο στοιχείο των τριών τελευταίων στοιχείων της λίστας L_{lower} .
6. Διάγραψε το πρώτο και το τελευταίο στοιχείο της λίστας L_{lower} για να μην βρίσκονται στα σημεία αυτά και στις δύο λίστες.
7. Συνένωσε τις λίστες L_{lower} και L_{upper} στη λίστα L .



Σχήμα 3: Εκτέλεση του αλγορίθμου του Andrews κατά την οποία διαγράφονται αρκετά ενδιάμεσα σημεία

Επίστρεψε τη λίστα L .

Αυτό που γίνεται λοιπόν σε πρώτη φάση είναι μία ταξινόμηση των σημείων μας βάσει της τετμημένης τους, ώστε τελικά το p_1 έχει τη μικρότερη τετμημένη και το p_n τη μεγαλύτερη. Κατόπιν, προσθέτουμε όλο και δεξιότερα σημεία (δηλαδή σημεία με μεγαλύτερες τετμημένες) στη λίστα μας και βάσει της ιδέας του αλγορίθμου κοιτάζουμε πότε γίνεται και πότε δεν γίνεται δεξιά στροφή. Αν δεν γίνεται, τότε διαγράφουμε τη μεσαία από τις τρεις τελευταίες κορυφές. Το πρακτικό αποτέλεσμα αυτής της ενέργειας είναι να αφαιρούμε αυτό το σημείο από το κυρτό περίβλημα μαζί με τις δύο πλευρές στις οποίες συμμετείχε, αφήνοντας τα δύο γειτονικά του σημεία μαζί με μία πλευρά που σχηματίζουν αυτά μεταξύ τους. Ένα παράδειγμα αυτής της διαδικασίας φαίνεται στο Σχήμα 3, όπου αρκετές ενδιάμεσες κορυφές φεύγουν από το τρέχον κυρτό περίβλημα. Αντίστοιχες διαδικασίες γίνονται και κατά το σχηματισμό του κάτω μέρους του κυρτού περιβλήματος.

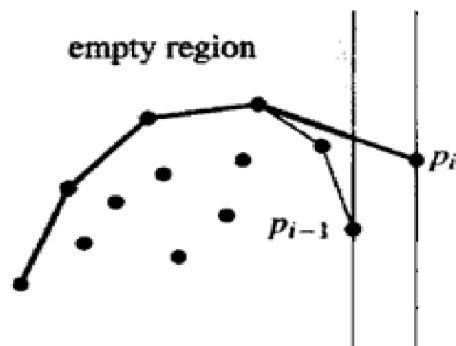
Ο αλγόριθμος έχει αρκετά καλή απόδοση, αλλά για να λειτουργήσει κάτω από πραγματικά δεδομένα χρειάζεται ορισμένες μικρές τροποποιήσεις για να μπορεί να αντιμετωπίσει ορισμένες ειδικές περιπτώσεις όπως είναι οι ακόλουθες:

- **Σημεία με ίδια τετμημένη:** προτείνεται να διατάσσονται τα σημεία πρώτα ως προς την τετμημένη και μετά ως προς την τεταγμένη (λεξικογραφική διάταξη).

- **Τρία σημεία επί ευθείας:** προτείνεται να θεωρείται ότι πρόκειται για αριστερή στροφή, ώστε να κρατούνται στο κυρτό περίβλημα μόνο τα δύο ακραία σημεία κατά τις διαγραφές σημείων.
- **Σφάλματα στρογγυλοποίησης:** είναι πιθανό κάποια από τις κορυφές που ανήκει στο κυρτό περίβλημα, λόγω στρογγυλοποιήσεων να διαγραφεί. Παρ' όλα αυτά, το τελικό αποτέλεσμα που θα πάρουμε επιστρέφει με πάρα πολύ καλή ακρίβεια το σωστό περίβλημα.

3.4 Απόδειξη της ορθότητας του αλγόριθμου του Andrews

Εδώ θα αποδείξουμε την ορθότητα της κατασκευής του άνω τμήματος του κυρτού περιβλήματος. Αντίστοιχα επιχειρήματα μπορούν να χρησιμοποιηθούν για να αποδειχθεί και η ορθότητα της κατασκευής του κάτω τμήματος. Η απόδειξη θα γίνει με επαγωγή ως προς τον αριθμό του στοιχείου που χειριζόμαστε την εκάστοτε στιγμή.



Σχήμα 4: Απόδειξη της ορθότητας του αλγορίθμου του Andrews

Πριν την εκτέλεση του βρόχου που ξεκινά με την εντολή 3 του ψευτοκώδικα έχουμε στη λίστα L_{upper} τα σημεία p_1 και p_2 , που σχηματίζουν το πάνω μέρος του κυρτού περιβλήματος των δύο αυτών σημείων. Ας υποθέσουμε τώρα ότι το L_{upper} περιέχει το άνω τμήμα του κυρτού περιβλήματος για τα σημεία p_1, p_2, \dots, p_{i-1} και ότι προσθέτουμε το σημείο p_i στη λίστα. Μετά από την εκτέλεση του βρόχου που ξεκινά με την

εντολή 5 του ψευτοκώδικά μας γνωρίζουμε ότι τα σημεία του L_{upper} σχηματίζουν μία αλυσίδα που περιέχει μόνο πλευρές που εκτελούν δεξιές στροφές. Επίσης, η αλυσίδα αυτή ξεκινά από το p_1 και τερματίζει στο p_i , λόγω της ταξινόμησης που είχαμε κάνει στο πρώτο βήμα του ψευτοκώδικά μας. Αν μπορέσουμε λοιπόν να δείξουμε ότι όλα τα σημεία p_1, p_2, \dots, p_i βρίσκονται κάτω από την αλυσίδα, τότε το L_{upper} περιέχει τα σωστά σημεία. Γνωρίζουμε ότι πριν από την προσθήκη του σημείου p_i κανένα σημείο δεν βρισκόταν πάνω από την αλυσίδα, από την επαγωγική υπόθεσή μας. Εφόσον η παλιά αλυσίδα βρίσκεται κάτω από την καινούρια, η μόνη πιθανότητα κάποιο σημείο να βρίσκεται πάνω από την καινούρια αλυσίδα είναι κάποιο σημείο να βρίσκεται μεταξύ του p_{i-1} και του p_i και πάνω από την αλυσίδα. Αυτό όμως δεν μπορεί να γίνει, καθώς τότε το σημείο αυτό (το οποίο θα είναι ένα εκ των p_1, p_2, \dots, p_{i-2}) θα είχε ταξινομηθεί λάθος στο πρώτο βήμα του ψευτοκώδικά μας. Αυτό φαίνεται και στο Σχήμα 4.

3.5 Πολυπλοκότητα αλγορίθμου

Όσον αφορά στη χρονική πολυπλοκότητα του αλγορίθμου του Andrews, η ταξινόμηση που γίνεται με το πρώτο βήμα του ψευτοκώδικά μας χρειάζεται κατά τα γνωστά χρόνο $O(n \log n)$. Τώρα, ας δούμε τις χρονικές απαιτήσεις που έχει η κατασκευή του άνω τμήματος του κυρτού περιβλήματος. Ο βρόχος που ξεκινά στην εντολή 3 του ψευτοκώδικα εκτελείται $\Theta(n)$ φορές. Το σημαντικό λοιπόν πλέον είναι να κατανοήσουμε πόσες φορές εκτελείται ο βρόχος που ξεκινά με την εντολή 5 του ψευτοκώδικά μας. Αυτός ο βρόχος εκτελείται τουλάχιστον μία φορά με κάθε εκτέλεση του εξωτερικού του βρόχου. Επίσης, για κάθε επιπλέον φορά που εκτελείται διαγράφει ένα σημείο από το τρέχον κυρτό περίβλημα. Προφανώς, κάθε στοιχείο μπορεί να διαγραφεί μία μόνο φορά. Επομένως, ο αριθμός των επιπλέον εκτελέσεων του εσωτερικού βρόχου φράσσεται από το n . Άρα, το συνολικό κόστος εκτέλεσης του άνω τμήματος του εξωτερικού βρόχου είναι $O(n)$ και ομοίως και του κάτω τμήματος. Άρα συνολικά το χρονικό κόστος του αλγορίθμου είναι $O(n \log n)$ και καθορίζεται πρακτικά από το κόστος που έχει η ταξινόμηση των σημείων.

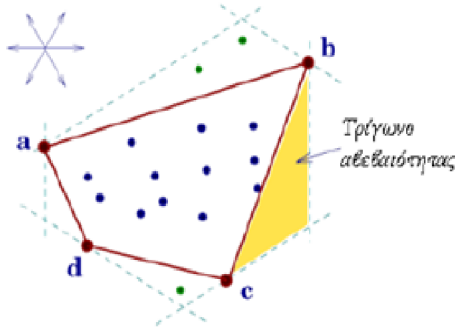
4 Προσεγγιστικός υπολογισμός κυρτού περιβλήματος ρεύματος σημειακών δεδομένων

Όπως αναφέραμε και στην ενότητα 2 του άρθρου μας, δεν είναι πρακτικά εφικτό να χρησιμοποιούμε αλγόριθμους που μας δίνουν το ακριβές κυρτό περίβλημα. Έτσι, επιλέγουμε να χρησιμοποιήσουμε προσεγγιστικούς αλγόριθμους, οι οποίοι δίνουν ένα περίβλημα αρκετά κοντά στο ζητούμενο, ή έστω ένα περίβλημα του οποίου το σφάλμα (όπως αυτό θα οριστεί στα ακόλουθα) από το ακριβές κυρτό περίβλημα είναι γνωστό ή έστω γνωρίζουμε μέχρι ποια τιμή μπορεί να κινείται, έχουμε δηλαδή ένα άνω φράγμα του σφάλματος.

Μάλιστα, προσεγγιστικούς αλγόριθμους υπολογισμού του κυρτού περιβλήματος χρησιμοποιούμε κυρίως για τα ρεύματα σημειακών δεδομένων για τα οποία θέλουμε να έχουμε σε πραγματικό χρόνο μία καλή προσέγγιση του κυρτού περιβλήματός τους. Τα ρεύματα δεδομένων, όπως προαναφέραμε, έχουν ιδιαίτερες απαιτήσεις. Ο όγκος δεδομένων μπορεί να είναι εξαιρετικά μεγάλος (εώς και απεριόριστος), ο ρυθμός με τον οποίο έρχονται τα δεδομένα μπορεί να είναι κυμαινόμενος, ενώ παράλληλα να έχουμε πολλά διαρκή ερωτήματα τα οποία τρέχουν διαρκώς. Έτσι, προτιμούμε έναν προσεγγιστικό αλγόριθμο ενός περάσματος για τέτοιου είδους δεδομένα.

Ο αλγόριθμος που θα χρησιμοποιήσουμε είναι αυτός που προτάθηκε από τους Hershberger και Suri [2]. Ο αλγόριθμος αυτός παίρνει ένα—ένα τα σημεία από το γεωμετρικό ρεύμα δεδομένων, το επεξεργάζεται και τέλος χωρίς να το αποθηκεύει έχει ένα επίκαιρο προσεγγιστικό κυρτό περίβλημα ανά πάσα στιγμή. Το κυρτό περίβλημα που δίνει ως έξοδο έχει κατά μέγιστο r κορυφές, όπου r είναι ένας ακέραιος που επιλέγει ο χρήστης. Επίσης, το αποτέλεσμα του αλγορίθμου είναι ένα πολύγωνο που περικλείεται εξ ολοκλήρου εντός του ακριβούς κυρτού περιβλήματος, ενώ το σφάλμα ορίζεται ως η μέγιστη απόσταση ανάμεσα στα δύο περιβλήματα.

Η βασική ιδέα από την οποία προήλθε ο ευριστικός αυτός αλγόριθμος είναι ότι κάθε κορυφή του ακριβούς κυρτού περιβλήματος είναι ακραία ως προς κάποια κατεύθυνση σε σχέση με όλα τα υπόλοιπα δεδομένα του



Σχήμα 5: Επίδειξη των τριγώνων αβεβαιότητας

συνόλου που εξετάζουμε. Έτσι, μπορούμε να βάλουμε εξ αρχής r καθορισμένες εκ των προτέρων κατευθύνσεις στις δύο διαστάσεις και να αναζητήσουμε τα πιο ακραία σημεία ως προς αυτές τις κατευθύνσεις. Αυτό κάνει και ο αλγόριθμος που εξετάζουμε.

Αν υποθέσουμε ότι τα σημεία p και q είναι διαδοχικές κορυφές του προσεγγιστικού κυρτού περιβλήματος που δίνει ο ευριστικός αλγόριθμος, τότε στο ακριβές κυρτό περίβλημα υπάρχει μία αλυσίδα που έχει συμπτυχθεί στο \overline{pq} . Για αυτό το λόγο και ορίζουμε ως σφάλμα από την πλευρά \overline{pq} τη μέγιστη απόσταση από το \overline{pq} προς τις κορυφές της αλυσίδας που ενώνει τα p και q στο πραγματικό κυρτό περίβλημα. Αυτό το σφάλμα δεν μπορεί να μετρηθεί ακριβώς, μπορεί όμως να φραγεί από το $\frac{l(\overline{pq}) \times \theta(\overline{pq})}{2}$, όπου $l(\overline{pq})$ είναι το μήκος του διανύσματος \overline{pq} και $\theta(\overline{pq})$ είναι η διαφορά των γωνιών των κατευθύνσεων στις οποίες είναι τα πιο ακραία σημεία τα p και q αντίστοιχα. Με τη βοήθεια των τριγώνων αβεβαιότητας (βλέπετε Σχήμα 5). Τα τρίγωνα αβεβαιότητας σχηματίζονται μεταξύ μίας πλευράς \overline{pq} του προσεγγιστικού κυρτού περιβλήματος και δύο γραμμών στήριξης, οι οποίες είναι γραμμές κάθετες προς τις κατευθύνσεις στις οποίες τα p και q είναι τα πιο απομακρυσμένα, και διέρχονται αντίστοιχα από τα p και q . Το προσεγγιστικό περίβλημα μαζί με τα τρίγωνα αβεβαιότητας περιλαμβάνουν πάντοτε ολόκληρο το πραγματικό κυρτό περίβλημα, για αυτό και μπορούμε να φράξουμε το σφάλμα με τη βοήθειά τους.

4.1 Περιγραφή του ευριστικού αλγορίθμου

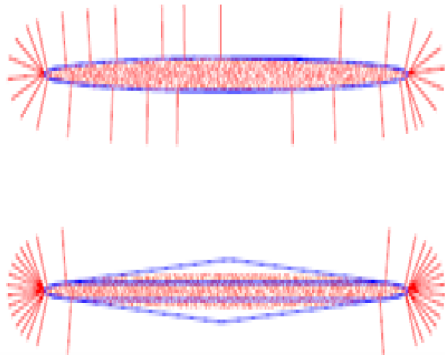
Ο αλγόριθμος προκύπτει εύκολα από τα προαναφερθέντα και είναι ο εξής.

Κατ' αρχάς επιλέγονται οι κατευθύνσεις επί των οποίων θα αναζητούμε το πιο ακραίο σημείο. Οι κατευθύνσεις αυτές είναι σταθερές κατά τη διάρκεια της εκτέλεσης του αλγορίθμου και ομοιόμορφα κατανομημένες πάνω στο μοναδιαίο κύκλο γι' αυτό και η δειγματοληψία χαρακτηρίζεται ομοιόμορφη. Η επιλογή των κατευθύνσεων είναι το εύκολο κομμάτι του αλγορίθμου, τουλάχιστον στην περίπτωση της ομοιόμορφης δειγματοληψίας που θα ακολουθήσουμε σε αυτό το άρθρο. Αν ορίσουμε $\theta_0 = 2\pi/r$, τότε οι επιθυμητές κατευθύνσεις είναι οι $j\theta_0$, για $j = 0, 1, 2, \dots, r-1$.

Τώρα ας δούμε τα επόμενα βήματα. Διαθέτουμε μία δομή, στην οποία αποθηκεύουμε κάθε φορά ποιο σημείο είναι πιο ακραίο ως προς κάθε μία από τις διευθύνσεις $j\theta_0$. Έτσι, όταν έρχεται το πρώτο σημείο, αυτό είναι το πιο ακραίο ως προς όλες τις διευθύνσεις. Έπειτα, όσο έρχονται σημεία από το ρεύμα δεδομένων, κάνουμε την ακόλουθη διαδικασία. Κοιτάζουμε αν είναι πιο ακραίο από το μέχρι πρότινος πιο ακραίο σημείο ως προς κάθε μία από τις κατευθύνσεις που επιλέξαμε, οπότε το αντικαθιστά. Διαφορετικά το μέχρι πρότινος ακραίο ως προς αυτήν την κατεύθυνση, παραμένει. Το κόστος για κάθε νέο σημείο που έρχεται είναι $O(r)$, διότι συγκρίνεται με r ακριβώς σημεία (αφού τόσες είναι και οι κατευθύνσεις που εξετάζουμε). Προφανώς $O(r) = O(1)$ αν θεωρήσουμε το r μία σταθερά, οπότε για ένα ρεύμα δεδομένων το οποίο δίνει n σημεία το συνολικό κόστος είναι γραμμικό ως προς n (δηλαδή $O(n)$).

4.2 Προσαρμοσμένη δειγματοληψία έναντι ομοιόμορφης

Όπως είδαμε στην περιγραφή του ευριστικού αλγορίθμου, οι κατευθύνσεις επιλέγονται ομοιόμορφα γύρω από το μοναδιαίο κύκλο. Όμως αυτή δεν είναι η μόνη μέθοδος που μπορούμε να ακολουθήσουμε. Μπορούμε να έχουμε εκτός από αυτές τις κατευθύνσεις και κάποιες άλλες. Συγκεκριμένα, αν ακολουθήσουμε τη μέθοδο της προσαρμοζόμενης δειγματοληψίας, τότε εκτός από τις r σταθερές κατευθύνσεις που θα έχουμε και θα είναι οι ίδιες με αυτές που έχουμε και στην ομοιόμορφη



Σχήμα 6: Η προσαρμοζόμενη δειγματοληψία (πάνω) και η ομοιόμορφη (κάτω). Είναι εμφανές ότι η προσαρμοζόμενη δίνει αρκετά καλύτερα αποτελέσματα

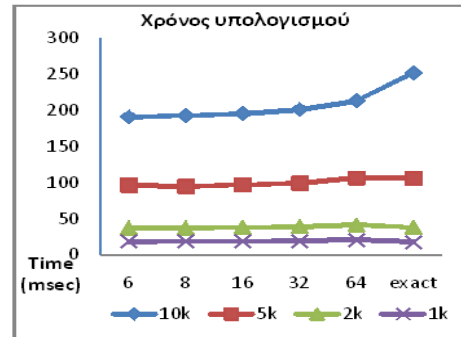
δειγματοληψία, θα μπορούμε να έχουμε ακόμα και $r + 1$ επιπλέον, οι οποίες θα προκύπτουν ανάλογα με τα δεδομένα τα οποία θα έχουμε. Έτσι μπορούμε να προσεγγίσουμε καλύτερα το πραγματικό κυρτό περιβλήμα και να έχουμε καλύτερα αποτελέσματα [2]. Όπως φαίνεται και στο Σχήμα 6 η προσαρμοζόμενη δειγματοληψία δίνει αρκετά καλύτερα αποτελέσματα.

5 Μελέτη επιδόσεων

5.1 Περιγραφή πειραμάτων

Η υλοποίηση των αλγορίθμων έγινε σε γλώσσα Java (έκδοση jre1.5.0.10 του java runtime environment). Η ανάπτυξη και εκτέλεση της εφαρμογής των πειραμάτων έγινε στην πλατφόρμα Eclipse έκδοση 3.2.1 σε λειτουργικό Windows[®] XP. Ο υπολογιστής που χρησιμοποιήθηκε για την εκτέλεση των πειραμάτων έχει επεξεργαστή Intel[®] Pentium[®] M συχνότητας 1.70 GHz και 1GB μνήμης RAM συχνότητας 590MHz. Για τη μέτρηση του χρόνου εκτέλεσης χρησιμοποιήθηκε η στατική μέθοδος της κλάσης System currentTimeMillis.

Ως είσοδο των αλγορίθμων χρησιμοποιήσαμε γεωμετρικά ρεύματα δεδομένων. Το κύριο σύνολο δεδομένων αποτελείται από 10,000 (10k) σημεία για 200 χρονικές στιγμές. Το σύνολο δεδομένων έχει τη μορφή $\langle T, ID, x, y \rangle$ για κάθε σημείο όπου T η χρονική



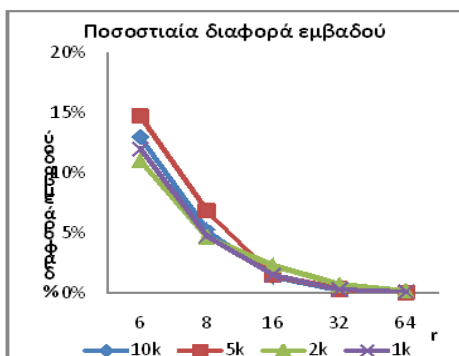
Σχήμα 7: Χρόνος υπολογισμού κυρτού περιβλήματος

στιγμή (timestamp), ID η ταυτότητα του κινούμενου αντικειμένου και (x, y) οι συντεταγμένες του. Από αυτό το κύριο σύνολο δεδομένων επιλέγοντας σημεία δημιουργήσαμε δευτερεύοντα σύνολα με μικρότερο μέγεθος. Η επιλογή των σημείων έγινε με βάση την τιμή του υπολοίπου του ID κάθε σημείου από τη διαίρεση του με το 10, 5, 2 και παρήχθησαν άλλα τέσσερα υποσύνολα με μέγεθος 1,000 (1k), 2,000 (2k) και 5,000 (5k) σημεία για κάθε χρονική στιγμή.

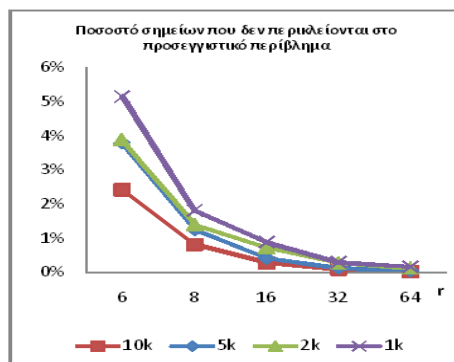
5.2 Μετρήσεις

Σκοπός των μετρήσεων είναι η σύγκριση της ταχύτητας των αλγορίθμων του Andrews και του ευριστικού και η μελέτη της επίδρασης του πλήθους των κατευθύνσεων που χρησιμοποιούνται από τον ευριστικό αλγόριθμο στην αποτελεσματικότητά του και στο χρόνο εκτέλεσής του.

Σε πρώτη φάση, θελήσαμε να μελετήσουμε τη διαφορά που υπάρχει στο χρόνο εκτέλεσης μεταξύ του ευριστικού αλγορίθμου για διάφορα r και του ακριβούς αλγορίθμου (ο οποίος στα ακόλουθα διαγράμματα απεικονίζεται ως exact). Για το σκοπό αυτό συγκρίνουμε αρχικά το χρόνο εκτέλεσης του αλγορίθμου του Andrews και το χρόνο εκτέλεσης του ευριστικού αλγορίθμου για αριθμό κατευθύνσεων $r = 6, 8, 16, 32, 64$ και για σύνολο δεδομένων μεγέθους 1k, 2k, 5k και 10k σημείων. Οι χρόνοι που εμφανίζονται στα αποτελέσματα αναφέρονται σε μέσο όρο χρόνου υπολογισμού για όλα τα χρονόσημα.



Σχήμα 8: Ποσοστιαία διαφορά του εμβαδού του περιβλήματος που δίνει ο ευριστικός αλγόριθμος σε σχέση με το ακριβές περίβλημα



Σχήμα 9: Ποσοστό των σημείων που δεν περιλαμβάνονται στο προσεγγιστικό περίβλημα

Από το Σχήμα 7, παρατηρούμε σε γενικές γραμμές ότι, για datasets μικρότερα ή ίσα από 5k δεν έχουμε ιδιαίτερες διαφοροποιήσεις στο χρόνο εκτέλεσης των αλγορίθμων. Όμως, στο dataset των 10k, βλέπουμε ότι όσο αυξάνεται το r για τον ευριστικό αλγόριθμο, τόσο αυξάνονται και οι χρονικές απαιτήσεις του αλγορίθμου. Επίσης, μπορούμε να παρατηρήσουμε ότι ο ακριβής αλγόριθμος του Ανδρεως κοστίζει αρκετά περισσότερο από τον προσεγγιστικό μας αλγόριθμο, ακόμα και όταν αυτός είχε παράμετρο $r = 64$. Συμπερασματικά, αύξηση του r οδηγεί σε αύξηση του κόστους του ευριστικού αλγορίθμου, ωστόσο το κόστος αυτό ακόμα και για μεγάλο r παραμένει μικρότερο από το αντίστοιχο του αλγορίθμου του Andrews.

Το ότι ο ευριστικός αλγόριθμος έχει μικρότερο κόστος από τον αλγόριθμο του Andrews είναι προφανές από την ανάλυση πολυπλοκότητας. Τώρα η αύξηση του κόστους του ευριστικού αλγορίθμου όσο αυξάνεται το r έχει επίσης την εξήγησή του. Όπως είχαμε δει στην ενότητα 4.1, το κόστος της 'επεξεργασίας' του κάθε νέου σημείου που έρχεται και εξετάζεται από τον αλγόριθμό μας είναι $O(r)$. Όσο αυξάνει το r , αναμένουμε σε γενικές γραμμές και μία αύξηση (γραμμική) του κόστους του αλγορίθμου. Όμως για ογκώδη σύνολα κινούμενων αντικειμένων είναι φανερό ότι ο ακριβής αλγόριθμος δεν μπορεί να θεωρηθεί αποδοτικός.

Στη συνέχεια, επιλέγουμε ως μέτρο σύγκρισης της αποτελεσματικότητας του ευριστικού αλγορίθμου για

διαφορετικό αριθμό κατευθύνσεων δύο μεγέθη: την ποσοστιαία διαφορά του εμβαδού του ακριβούς περιβλήματος των σημείων από το εμβαδό του ευριστικού και το πλήθος των σημείων του dataset τα οποία δεν περιλαμβάνονται στο περίβλημα που παράγει ο ευριστικός αλγόριθμος.

Η ποσοστιαία διαφορά εμβαδού ανάμεσα στο ακριβές κυρτό περίβλημα και στην προσέγγισή του, που παράγει ο ευριστικός αλγόριθμος, για όλα τα μεγέθη του dataset εισόδου σχεδόν μηδενίζεται για $r = 64$ και $r = 32$, ενώ ήδη για $r = 16$ η προσέγγιση είναι κοντά ή κάτω από το 2% για όλα τα σύνολα δεδομένων που δοκιμάστηκαν.

Όπως και προηγουμένως, από το Διάγραμμα 3 επαληθεύεται το συμπέρασμα ότι για πλήθος διευθύνσεων $r = 32$ ή 64 προσεγγίζουμε σε πολύ μεγάλο βαθμό την ακριβή λύση. Ακόμη και για μικρότερο r (για παράδειγμα 8 ή 16) τα σημεία που δεν περιλαμβάνονται στο περίβλημα είναι λιγότερα από το 2% των συνολικών σημείων για όλα τα σύνολα δεδομένων που ελέγξαμε.

Ας εξηγήσουμε τώρα κάπως θεωρητικά την ακρίβεια του κυρτού περιβλήματος που μας δίνει ο ευριστικός αλγόριθμος ως προς το ακριβές κυρτό περίβλημα. Κατ' αρχάς από το Διάγραμμα 2 φαίνεται ότι όσο αυξάνεται η παράμετρος r , τόσο πιο ακριβές είναι το κυρτό περίβλημα του ευριστικού μας αλγορίθμου. Αυτό είναι λογικό, διότι αναμέναμε πως όσες περισσότερες κατευθύνσεις έχουμε για να αναζητήσουμε ακραία σημεία, τόσο καλύτερες προσεγγίσεις θα είχαμε. Μπορούμε να πούμε

επίσης, ότι ο ακριβής αλγόριθμος θεωρητικά αναζητεί ως προς όλες τις δυνατές κατευθύνσεις τα ακραία σημεία. Με τον προσεγγιστικό αλγόριθμο, αναζητούμε τα ακραία σημεία ως προς πεπερασμένες κατευθύνσεις μόνο. Αν θελήσουμε για παράδειγμα να έχουμε $r = 4$, τότε θα προκύψει ως κυρτό περιβλήμα ένα τετράγωνο (ή και τρίγωνο αν ένα στοιχείο είναι το πιο ακραίο ως προς δύο κατευθύνσεις ή ακόμα και ευθύγραμμο τμήμα), αν έχουμε $r = 8$ θα προκύψει ένα οκτάγωνο (κατά μέγιστο) κ.ο.κ.. Αυτό σημαίνει, ότι όσο αυξάνουμε το r τόσο προσεγγίζουμε το πραγματικό κυρτό περιβλήμα. Μάλιστα, για $r = \infty$ θεωρητικά θα παίρναμε το ακριβές κυρτό περιβλήμα, χωρίς κανένα σφάλμα.

Αντίστοιχα πράγματα μπορούμε να δούμε και από το Διάγραμμα 3. Όσο αυξάνεται το r και βελτιώνεται η προσέγγιση του κυρτού περιβλήματος τόσο μειώνεται και το πλήθος των σημείων που παραμένουν εκτός του κυρτού περιβλήματος που επιστρέφει ο προσεγγιστικός αλγόριθμος.

6 Συμπεράσματα

Στην παραπάνω εργασία εξετάσαμε το πρόβλημα εύρεσης, προσεγγιστικής ή ακριβούς, του κυρτού περιβλήματος είτε ενός στατικού συνόλου σημείων είτε ενός δυναμικού ρεύματος δεδομένων. Το πρόβλημα αυτό εντάσσεται σε μια ομάδα θεμελιωδών προβλημάτων της υπολογιστικής γεωμετρίας. Άλλα προβλήματα αυτής της ομάδας είναι ο υπολογισμός του διαγράμματος Voronoi και του τριγωνισμού Delaunay (περισσότερες λεπτομέρειες για αυτά και για άλλα προβλήματα της υπολογιστικής γεωμετρίας μπορούν να αναζητηθούν στο [7]). Η έρευνα γύρω από αυτά τα θεμελιώδη προβλήματα είναι έντονη ενώ η ανάγκη για μελέτη και διαχείριση ογκωδών συνόλων κινούμενων αντικειμένων επαναπροσανατολίζει και δίνει νέες παραμέτρους στους υπάρχοντες αλγόριθμους.

Όπως προέκυψε από τα πειράματα που εκτελέσαμε, καθώς και από τη θεωρητική τεκμηρίωση των αποτελεσμάτων που λάβαμε από τη διενέργεια των πειραμάτων, μπορούμε να πούμε ότι ο ευριστικός αλγόριθμος που μελετήθηκε δίνει εξαιρετικά ικανοποιητικά αποτελέσματα.

Κατ' αρχάς, το χρονικό του κόστος για μεγάλα datasets είναι αρκετά μικρότερο από το αντίστοιχο κό-

στος του αλγορίθμου του Andrews, που θεωρητικά είναι βέλτιστος αλγόριθμος υπολογισμού του ακριβούς κυρτού περιβλήματος. Στη συνέχεια, παρατηρήσαμε ότι με σωστή επιλογή του r μπορούμε να πάρουμε πολύ καλά αποτελέσματα και τέλος η χρήση του αλγορίθμου μας επιτρέπει να γνωρίζουμε ανά πάσα στιγμή μία τιμή κάτω από την οποία κινείται το σφάλμα της προσέγγισής μας. Για όλους τους ανωτέρω λόγους κρίνουμε ότι η χρησιμοποίηση του ευριστικού αλγορίθμου σε γεωμετρικά ρεύματα σημειακών δεδομένων δίνει πολύ καλά και αξιόπιστα αποτελέσματα, με μικρά και ελεγχόμενα σφάλματα, μικρό χρονικό κόστος και μικρή χρησιμοποίηση της μνήμης.

Αναφορές

- [1] J. Hershberger and S. Suri. Convex Hulls and Related Problems in Data Streams. In Proceedings of the ACM SIGMOD/PODS Workshop on Management and Processing of Data Streams, San Diego, California, USA, June 2003.
- [2] J. Hershberger and S. Suri. Adaptive Sampling for Geometric Problems over Data Streams. In Proceedings of the 23rd ACM Symposium on Principles of Database Systems (PODS'04), pp. 252-262, Paris, France, June 2004.
- [3] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. Computational Geometry: Algorithms and Applications. Springer-Verlag, Berlin, Germany, 2nd edition, 2000.
- [4] P. Rigaux, M. Scholl, and A. Voisard. Spatial Databases with Application to GIS, Morgan Kaufmann, 2001.
- [5] P. Indyk. Stream-based geometric algorithms. In Proceedings of the ACM SIGMOD/PODS Workshop on Management and Processing of Data Streams, San Diego, California, USA, June 2003.
- [6] P. Indyk. Algorithms for dynamic geometric problems over data streams. In Proceedings of the 36th ACM Symposium on Theory of Computing, Chicago, Illinois, USA, pp. 373-380, 2004.

- [7] Κ. Πατρούμπας, Γεωμετρικοί υπολογισμοί σε σημειακά δεδομένα, Ιούλιος 2005