

Δημιουργία αλγορίθμου και χαρτογραφική απόδοση του χρόνου μετακίνησης στο λεκανοπέδιο Αττικής με Δημόσια Μέσα Μεταφοράς

Περίληψη

Σκοπός της παρούσας εργασίας είναι να δημιουργηθούν χάρτες με καμπύλες χρονοαπόστασης στο δίκτυο Μέσων Μαζικής Μεταφοράς της Αθήνας με αφετηρία την πλατεία της Ομόνοιας. Στόχος είναι τα αποτελέσματα της ανάλυσης να είναι ρεαλιστικά και οι χρόνοι μετακίνησης λογικοί.

Ισοχρονικές καμπύλες χρονοαπόστασης καλούνται οι καμπύλες που ορίζουν ένα πολύγωνο πάνω στον χάρτη το οποίο περιλαμβάνει όλες τις περιοχές οι οποίες είναι προσβάσιμες μέσα σε ένα ορισμένο χρονικό διάστημα από ένα αρχικό σημείο. Ανάλογα με το δίκτυο στο οποίο γίνεται μία τέτοια ανάλυση αλλάζουν οι διαδικασίες και οι παράμετροι δημιουργίας των ισοχρονικών καμπυλών. Οι περισσότερες εργασίες που έχουν δημοσιευτεί μέχρι σήμερα αφορούν στα οδικά δίκτυα.

Στα πλαίσια της εργασίας αυτής αναπτύχθηκε πρωτότυπος κώδικας που επιλύει το πρόβλημα των ισοχρονικών καμπυλών για τα μέσα μαζικής μεταφοράς βασισμένος στον αλγόριθμο Dijkstra. Αντιμετωπίστηκαν οι ιδιαιτερότητες του συγκεκριμένου δικτύου όπως οι πολλαπλές ακμές ίδιου κόστους που μπορεί να περιέχονται ανάμεσα σε δύο κόμβους, το επιπρόσθετο κόστος που υπεισέρχεται σε περιπτώσεις αλλαγής κωδικού γραμμής (μετεπιβίβασης), το γεγονός ότι οι κόμβοι (στάσεις) δεν βρίσκονται επί του δικτύου αλλά παράλληλα σ' αυτό και άλλες που προέκυψαν στη διάρκεια της εργασίας.

Επιπλέον σχεδιάστηκε και υλοποιήθηκε μία διαδικτυακή χαρτογραφική εφαρμογή για την απεικόνιση των δημιουργημένων ισοχρονικών καμπυλών. Ο χρήστης έχει τη δυνατότητα να ορίσει το χρόνο μετακίνησης που επιθυμεί να απεικονίσει η ισοχρονική καμπύλη, καθώς και συμπληρωματικά θεματικά επίπεδα που συμπληρώνουν τον χάρτη όπως το χαρτογραφικό υπόβαθρο Open Street Map, το δίκτυο του Αττικό Μετρό και το δίκτυο των λεωφορειακών γραμμών με τις στάσεις τους.

Λέξεις Κλειδιά: ισοχρονικές καμπύλες, χρονοαπόσταση, συντομότερη διαδρομή, Dijkstra

Εισαγωγή

Οι τεχνολογικές εξελίξεις κατά τα τέλη του 20ου αιώνα έχουν μεταβάλλει άρδην τη φύση και τα προϊόντα της Χαρτογραφίας και των Συστημάτων Γεωγραφικών Πληροφοριών (GIS). Η ραγδαία ανάπτυξη και η ευρεία χρήση των ηλεκτρονικών υπολογιστών και του διαδικτύου, ιδιαίτερα του Παγκόσμιου Ιστού (Web), έχουν επηρεάσει σημαντικά και έχουν

καθορίσει νέα πρότυπα και προδιαγραφές σε όλα τα στάδια της επεξεργασίας και ανάλυσης γεωχωρικών δεδομένων και της χαρτογραφικής παραγωγής.

Οι διαδικτυακές χαρτογραφικές υπηρεσίες, τα συστήματα πλοήγησης και άλλες εφαρμογές σχεδιασμού διαδρομών και εντοπισμού θέσης έχουν αποκτήσει ευρεία χρήση λόγω της σημαντικής προόδου των αλγορίθμων συντομότερων διαδρομών.

Το πρόβλημα εύρεσης της συντομότερης διαδρομής (shortest path problem) σε δίκτυα οδικά, σιδηροδρομικά, δίκτυα κοινής ωφέλειας, δίκτυα τηλεπικοινωνιών κ.λπ., συνιστά ένα από τα θεμελιώδη προβλήματα της σύγχρονης επιστήμης των υπολογιστών και αφορά ένα πλήθος εφαρμογών πλοήγησης και σχεδιασμού δικτύων.

Αντικείμενο εργασίας

Τα προηγούμενα χρόνια, οι διαδραστικοί διαδικτυακοί χάρτες έγιναν ένα πολύ διαδεδομένο εργαλείο για τον σχεδιασμό διαδρομών όλων των ειδών. Σήμερα κάθε άνθρωπος με πρόσβαση στο διαδίκτυο μπορεί εύκολα να βρει τις διαθέσιμες διαδρομές όταν ταξιδεύει από ένα σημείο σε ένα άλλο. Οι ιστοσελίδες που προσφέρουν δρομολόγηση χρησιμοποιούν συνήθως αλγορίθμους συντομότερης διαδρομής (SP).

Λιγότερο γνωστό αλλά αλγοριθμικά πολύ ενδιαφέρον είναι να βρεθεί η απάντηση στο ως πού μπορεί κάποιος να ταξιδέψει σε δεδομένα χρόνο, ξεκινώντας από δεδομένο σημείο. Το αποτέλεσμα μιας τέτοιας ανάλυσης μπορεί να απεικονιστεί με ένα είδος ισαριθμικών καμπύλων που καλείται ισοχρονική καμπύλη.

Ένα από τα σημαντικότερα προτερήματα των ισοχρονικών καμπυλών είναι ότι μπορούν να χρησιμοποιηθούν για αναλύσεις προσβασιμότητας όλων των ειδών. Μπορούν να αποτελέσουν βοηθητικό εργαλείο σε ποικίλα πεδία, όπως στον αστικό σχεδιασμό και στη διαχείριση έκτακτων καταστάσεων.

Αντικείμενο της παρούσας εργασίας είναι η δημιουργία ενός αλγοριθμικού συστήματος για τον γρήγορο υπολογισμό ισοχρονικών καμπυλών στο δημόσιο δίκτυο μεταφορών της Αθήνας. Ιδιαίτερη έμφαση δίνεται στον υπολογισμό των χρονικά συντομότερων διαδρομών μεταξύ δύο σημείων του δικτύου, καθώς η διαδικασία αυτή έχει αρκετές ιδιαιτερότητες όταν αφορά δίκτυο μέσων μαζικής μεταφοράς. Επιπλέον ένα σημαντικό μέρος της εργασίας αποτέλεσε η διαδικασία οπτικοποίησης το εν λόγω καμπυλών και η παρουσίασή τους μέσω χαρτογραφικής διαδικτυακής εφαρμογής.

Θεωρητικό πλαίσιο

Ένας ισοχρονικός χάρτης στη Χαρτογραφία, είναι ένας χάρτης που απεικονίζει περιοχές που συνδέονται με μία ισοχρονική καμπύλη. Η ισοχρονική καμπύλη ορίζεται ως μια γραμμή σχεδιασμένη στο χάρτη με τρόπο ώστε να συνδέει σημεία στα οποία κάτι συμβαίνει ή φτάνει την ίδια στιγμή. Στο σχεδιασμό των μεταφορών, οι ισοχρονικοί χάρτες χρησιμοποιούνται για να απεικονίσουν περιοχές με ίση διάρκεια ταξιδιού από μία κοινή αφετηρία.

Το αντικείμενο της θεωρίας των γράφων είναι η μελέτη των γράφων, δηλαδή μαθηματικών δομών, που χρησιμοποιούνται για τη μοντελοποίηση και την αναπαράσταση των σχέσεων που υφίστανται μεταξύ ζευγών αντικειμένων. Σύμφωνα με τον ορισμό που προέρχεται από τα μαθηματικά, ένας γράφος $G = (V, E)$ συνιστά μια αναπαράσταση ενός συνόλου

αντικειμένων που συνδέονται μεταξύ τους. Τα αντικείμενα καλούνται κόμβοι ή κορυφές (Vertices) του γράφου και συμβολίζονται με V και οι συνδέσεις που υφίστανται ανάμεσα σε ένα ζεύγος κόμβων ονομάζονται ακμές (Edges) του γράφου και συμβολίζονται με E .

Τύποι Γράφων

Οι γράφοι εμφανίζονται διαφοροποιημένοι μεταξύ τους, ανάλογα με την ιδιαίτερη δομή και τα χαρακτηριστικά τους. Ως εκ τούτου, ταξινομούνται σε επιμέρους κατηγορίες στη βάση ενός αριθμού κριτηρίων που τίθενται κατά περίπτωση, όπως για παράδειγμα η ανάθεση ή μη φοράς στις ακμές του γράφου, η συνδεσιμότητα του γράφου, η ανάθεση βαρών στις ακμές του γράφου κ.λπ.

Ανάλογα με τη δομή και τα χαρακτηριστικά τους, οι γράφοι υιοθετούνται για τη μοντελοποίηση διαφορετικής φύσεως προβλημάτων. Ένας κατευθυνόμενος γράφος για παράδειγμα, συνιστά την καταλληλότερη δομή για τη μοντελοποίηση δικτύων ενώ ένας απλός μη κατευθυνόμενος γράφος, αποτελεί κατάλληλη δομή για την αναπαράσταση των χημικών δεσμών που υφίστανται μεταξύ των ατόμων ενός μορίου.

Αλγόριθμοι ελάχιστης διαδρομής

Οι αλγόριθμοι που έχουν αναπτυχθεί για τη δημιουργία καμπυλών ίσου χρόνου είναι βασισμένοι στον αλγόριθμο ελάχιστης διαδρομής του Dijkstra. Σαφώς έχουν αναπτυχθεί και άλλες προσεγγίσεις της εύρεσης της συντομότερης διαδρομής συνεπώς είναι απαραίτητη η μελέτη των βασικότερων από αυτές, ώστε να αποφασιστεί αν είναι χρήσιμες για την παρούσα εργασία.

Dijkstra

Πρόκειται για έναν αλγόριθμο εύρεσης των συντομότερων διαδρομών (single-source shortest path problem), από κοινή αφετηρία σε έναν (κατευθυνόμενο ή μη) γράφο με μη αρνητικά βάρη στις ακμές.

Συνεπώς δεδομένου ενός γράφου $G = (V, E)$, όπου V είναι το σύνολο των κόμβων και E το σύνολο των ακμών, και ενός αρχικού κόμβου s , ο αλγόριθμος Dijkstra υπολογίζει τις συντομότερες διαδρομές από το σημείο αυτό προς όλους τους κόμβους του γράφου (Langeman Matt, 2005).

Ο αλγόριθμος διατηρεί ένα πίνακα D στον οποίο αποθηκεύει την τρέχουσα υπολογισμένη ελάχιστη απόσταση κάθε κόμβου από τον αρχικό s . Κατά την αρχικοποίηση $D[s] = 0$ και $D[u] = +\infty$, όπου u είναι κάθε κόμβος διάφορος του s . Επίσης, διατηρεί ένα σύνολο S που αποτελείται από τους κόμβους για τους οποίους έχει ήδη προσδιοριστεί η ελάχιστη διαδρομή και το οποίο είναι αρχικά κενό. Τέλος, ο αλγόριθμος χρησιμοποιεί ένα ακόμη διάνυσμα, το $prev[]$, μεγέθους n , στο οποίο για κάθε κόμβο u αποθηκεύεται ο αμέσως προηγούμενος κόμβος στο ελάχιστο μονοπάτι προς τον u (Παπαθεοδώρου Σ. Θεόδωρος, 2007).

Για τον υπολογισμό των ισοχρονικών καμπυλών μπορεί και πάλι να χρησιμοποιηθεί ο αλγόριθμος Dijkstra χρησιμοποιώντας ως κόστος το χρόνο διαδρομής αντί για την απόσταση.

Ο αλγόριθμος A*

Ένας ακόμη αλγόριθμος ελάχιστης διαδρομής βασισμένος στον αντίστοιχο του Dijkstra είναι ο A*. Χρησιμοποιείται σε περιπτώσεις μεγάλων γράφων όπου η πολυπλοκότητα του αλγόριθμου του Dijkstra καθίσταται απαγορευτική. Αξιοποιεί τεχνικές της τεχνητής νοημοσύνης, προκειμένου να αυξηθεί η ταχύτητα αναζήτησης στους γράφους μέσα από την εφαρμογή ευρετικών κανόνων (heuristics).

Αρχικά, εκτιμάται το κόστος μετάβασης από τον κόμβο αφετηρίας στον κόμβο προορισμού. Πρόκειται για μια ενδεικτική και όχι για μια ακριβή εκτίμηση. Στη συνέχεια, υπολογίζονται αναδρομικά τα κόστη μετάβασης στους γειτονικούς κόμβους του αρχικού κόμβου και αποκλείονται εξ αρχής μονοπάτια που έχουν μεγαλύτερο κόστος σε σχέση με αυτό που είχε αρχικά εκτιμηθεί (Στεφανάκης Εμμανουήλ, 2003).

Ο αλγόριθμος Bellman –Ford

Ο αλγόριθμος Bellman-Ford ο οποίος εφαρμόζεται σε ένα γράφο με συσχετισμένους συντελεστές βάρους. Αρχικά εντοπίζει το ελάχιστο μονοπάτι από το κόμβο εκκίνησης s αποτελούμενο από μία ζεύξη, στη συνέχεια το ελάχιστο μονοπάτι από τον s αποτελούμενο από δύο ζεύξεις και συνεχίζει κατ' αυτόν τον τρόπο για $n-1$ φορές όπου n το πλήθος των κόμβων στο γράφο. Αρχικά όλοι κόμβοι θεωρούνται ότι έχουν άπειρη απόσταση από τον s . Τελικά υπολογίζει τη συντομότερη διαδρομή από ένα σημείο εκκίνησης προς όλους τους υπόλοιπους κόμβους του γράφου, είναι όμως πιο αργός από τον Dijkstra στην επίλυση του ίδιου προβλήματος. Από την άλλη αποτελεί πιο ευέλικτη λύση αφού παρέχεται η δυνατότητα να χειρίζεται γραφήματα στα οποία υπάρχουν ακμές αρνητικού κόστους (<https://en.wikipedia.org/wiki/bellman>).

Χαρακτηριστικά δεδομένων

Είναι απαραίτητο να δημιουργηθεί ένας κατευθυνόμενος γράφος με βάρη στις ακμές του που θα περιγράφει το δίκτυο των δημόσιων συγκοινωνιών της Αθήνας. Ο γράφος θα πρέπει να αποτελείται από τα εξής μέρη :

- Κόμβοι οι οποίοι αντιπροσωπεύουν τα σημεία όπου υπάρχουν στάσεις. Οι κόμβοι θα πρέπει να περιγράφονται από τις γεωγραφικές συντεταγμένες τους.
- Ακμές οι οποίες αντιπροσωπεύουν τους άξονες πάνω στους οποίους κινούνται τα δημόσια μέσα μεταφοράς.
- Τα βάρη των ακμών τα οποία αντιπροσωπεύουν το χρόνο προσπέλασης κάθε ακμής.

Επιπλέον προστίθεται μία ακόμα παράμετρος για να ολοκληρωθεί η δημιουργία του γράφου του δικτύου, το οποίο είναι:

- Η συχνότητα κάθε δρομολογίου

Πολύ σημαντική είναι η πληροφορία της κατεύθυνσης των ακμών, η οποία δεν παρέχεται. Είναι αυτονόητο ότι κάθε γραμμή λεωφορείου/μετρό/τραμ μπορεί να κινηθεί μόνο προς μία κατεύθυνση, γεγονός που επηρεάζει καθοριστικά τη διαδικασία δρομολόγησης αφού εισάγει επιπλέον περιορισμούς.

Κατασκευή γράφου

Για την κατασκευή του γράφου χρησιμοποιήθηκε η βιβλιοθήκη pgRouting της Postgres. Μία από τις συναρτήσεις που παρέχει η εν λόγω βιβλιοθήκη είναι το `pgr_createTopology`. Με τη συνάρτηση αυτή ουσιαστικά χτίζεται η τοπολογία του γράφου μας βασισμένος στη γεωμετρία του δικτύου.

Η συνάρτηση επιστρέφει OK αν έχει κτιστεί η τοπολογία και FAIL αν υπήρξε κάποιο σφάλμα κατά τη διαδικασία. Επίσης ενημερώνει το χρήστη για τον αριθμό των κόμβων που δημιουργήθηκαν και τον αριθμό των εγγραφών που επηρεάστηκαν στον πίνακα των ακμών.

Επειδή στις μετακινήσεις με τα Δημόσια Μέσα Μεταφοράς, ένα σημαντικό στοιχείο είναι η σύνδεση των διαφορετικών μέσων και η μετεπιβίβαση, έπρεπε να δημιουργηθούν μονοπάτια περπατήματος για τους επιβάτες. Όπως και σε κάποιες διαδικτυακές εφαρμογές έτσι και εδώ συνδέθηκαν οι κόμβοι του δικτύου με ευθείες γραμμές και υπολογίστηκε η ευκλείδεια απόστασή μεταξύ τους.

Έλεγχος και διορθώσεις γράφου

Η βιβλιοθήκη PgRouting παρέχει κάποιες συναρτήσεις για εφαρμογή ελέγχων και διορθώσεων στο δίκτυο. Χρησιμοποιήθηκαν κάποιες από αυτές για να αξιολογηθεί ο γράφος που δημιουργήθηκε αλλά και για να εξαχθούν τα απαραίτητα στατιστικά στοιχεία.

Πιο συγκεκριμένα, αρχικά εφαρμόστηκε η συνάρτηση `pgr_analyzegraph`, που ελέγχει την τοπολογία του γράφου. Εντοπίζει τα αδιέξοδα (dead ends), τα απομονωμένα τμήματα δικτύου (isolated segments), τμήματα δηλαδή που και τα δύο άκρα είναι αδιέξοδα, τα τμήματα που έχουν γεωμετρία δακτυλίου (ring geometries), ενδεχόμενα κενά κοντά σε αδιέξοδα και τις διασταυρώσεις ακμών.

Η συνάρτηση επιστρέφει το μήνυμα OK αφού ολοκληρωθεί η ανάλυση.

Για να διορθωθούν τα σφάλματα αυτά, το PgRouting παρέχει τη συνάρτηση `pgr_nodeNetwork` η οποία έχει σχεδιαστεί για να διορθώνει την έλλειψη κόμβων στις διασταυρώσεις. Παρ' όλο που ο σκοπός του είναι η εισαγωγή κόμβων στις διασταυρώσεις τελικά διορθώνονται και άλλα προβλήματα όπως τα αδιέξοδα και τα κενά κοντά στο τέρμα των ακμών. Επειδή στην περίπτωση της εργασίας οι ακμές ορίζουν τη διαδρομή ενός μέσου από στάση σε στάση δεν μπορεί να αξιοποιηθεί η συνάρτηση αυτή, καθώς δεν πρέπει να εισαχθούν νέοι κόμβοι. Οι διορθώσεις στις προβληματικές περιοχές έγιναν οπτικά στο QGIS.

Μία ακόμη συνάρτηση ελέγχου που εντοπίζει ζητήματα κατεύθυνσης είναι η `pgr_analyzeOneway`. Αναλύει τις ακμές ενός γράφου σύμφωνα με τους κανόνες κατεύθυνσης που έχουν τεθεί και επισημαίνει εκείνες που τους παραβιάζουν.

Απόδοση κόστους στις ακμές

Ο γράφος ενός οδικού δικτύου ορίζεται ως το σύνολο των σημείων και των ακμών που τα συνδέουν, καθώς και το κόστος που αποδίδεται στις ακμές και αντιπροσωπεύει το κόστος (χιλιομετρικό, χρονικό, οικονομικό κ.λπ.) που απαιτείται για να μεταβεί κάποιος από κόμβο σε κόμβο.

Στο `pgrRouting` υπάρχουν δύο είδη κόστους. Το απλό κόστος που αναφέρεται στο κόστος διαδρομής από τον κόμβο αφετηρίας στον κόμβο προορισμού και το αντίστροφο κόστος το οποίο υποδηλώνει το κόστος πορείας αντίθετης φοράς από τη φορά σχεδίασης. Επειδή κάθε ακμή του γράφου είναι μονής κατεύθυνσης μπορεί να αγνοηθεί το αντίστροφο κόστος και στους υπολογισμούς ο γράφος να θεωρηθεί κατευθυνόμενος.

Σε αυτό το στάδιο πρέπει να υπολογιστεί το κόστος κάθε ακμής. Αφού το ενδιαφέρον εστιάζεται στον ελάχιστο χρόνο διαδρομής, το κόστος αρχικά θα είναι ο χρόνος προσπέλασης κάθε ακμής. Η διαδικασία αυτή έγινε στο περιβάλλον `PgAdmin` με χρήση της `Postgres`.

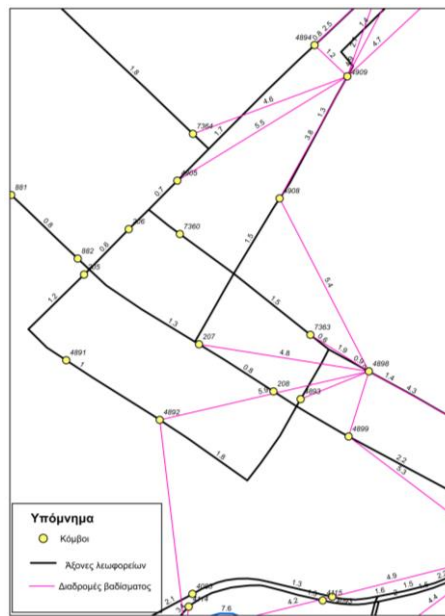
Για να αποδοθεί χρονικό κόστος σε κάθε γραμμή θα πρέπει να είναι γνωστή η ταχύτητα κάθε μέσου. Από τον αντίστοιχο πίνακα εξάγεται το συνολικό μήκος που καλύπτει κάθε ένα και τη διάρκεια κάθε διαδρομής από την αφετηρία μέχρι το τέρμα και με την εξίσωση

$$u_{\mu} = \frac{x}{t}, \text{ υπολογίζεται η μέση ταχύτητα σε μέτρα/λεπτό.}$$

Αφού υπολογιστεί και το μήκος κάθε ακμής με την αντίστοιχη συνάρτη `ST_length` που

προσφέρει η επέκταση `PostGis` της `Postgres`, εφαρμόζεται και πάλι η εξίσωση $t = \frac{x}{u_{\mu}}$ και υπολογίζεται ο χρόνος που χρειάζεται για να διασχιστεί κάθε ακμή.

Το γεγονός ότι πρόκειται για δρομολόγηση σε δημόσια μέσα μεταφοράς, εισάγει ένα επιπλέον χρονικό κόστος στα δεδομένα μας, το κόστος μετεπιβίβασης. Το κόστος αυτό, δηλαδή ο χρόνος αναμονής για μετάβαση σε επόμενο μέσο στην ίδια διαδρομή, εξαρτάται από τη συχνότητα του εν λόγω μέσου. Συνεπώς δημιουργήθηκε μία νέα στήλη με το όνομα `freq` η οποία θα συμπληρωθεί με τη συχνότητα κάθε γραμμής.



Εικόνα 1 : Απόδοση κόστους στις ακμές

Ανάπτυξη πρωτότυπου αλγορίθμου

Ο κορμός της εφαρμογής δεν είναι άλλος από τον υπολογισμό των σημείων του οδικού δικτύου στα οποία μπορεί να φτάσει ένας επιβάτης μέσα στο ζητούμενο χρονικό διάστημα ξεκινώντας από το αρχικό σημείο. Για τον υπολογισμό αυτό καταλληλότερος είναι ο αλγόριθμος συντομότερων μονοπατιών Dijkstra. Βέβαια, για τις ανάγκες της εφαρμογής χρειάστηκαν κάποιες βελτιώσεις μετά την εφαρμογή του αλγορίθμου. Σε αυτήν την παράγραφο, περιγράφεται η εφαρμογή του Dijkstra και αυτές οι απαιτούμενες βελτιώσεις.

Θέματα προς επίλυση

Ο αλγόριθμος Dijkstra υπολογίζει τα συντομότερα μονοπάτια. Για τον υπολογισμό ισοχρονικών καμπυλών όμως, το ενδιαφέρον εστιάζεται στους συντομότερους χρόνους διαδρομής. Το πρόβλημα αυτό ξεπερνιέται εύκολα αν ως κόστος οριστεί ο χρόνος διαδρομής και όχι η χιλιομετρική απόσταση.

Επιπλέον στην εύρεση της χρονικά συντομότερης διαδρομής με μέσα μαζικής μεταφοράς, πρέπει να ληφθεί υπ' όψιν ότι η μετεπιβίβαση επιφέρει ένα επιπλέον χρονικό κόστος. Το κόστος αυτό θα πρέπει να υπολογιστεί στην ανάλυση. Επίσης είναι αναγκαίο, το σύστημα να εισάγει μετεπιβίβαση μόνον όταν μειώνεται το συνολικό κόστος διαδρομής ή όταν δεν υπάρχει εναλλακτική.

Εφαρμόζοντας τον Dijkstra ανάμεσα σε ζεύγη κόμβων του γράφου, έγινε φανερό ότι υπολογίζονται μεν τα συντομότερα μονοπάτια, εισάγονται δε άσκοπες μετεπιβιβάσεις, γεγονός που καθιστά ανέφικτο τον σωστό υπολογισμό του χρονικού κόστους κάθε κόμβου. Αυτό οφείλεται στο γεγονός ότι στο γράφο του δικτύου ανάμεσα σε δύο κόμβους μπορεί να περιλαμβάνονται περισσότερες από μία ακμές με το ίδιο κόστος. Δηλαδή περισσότερες από μία γραμμές συνδέουν τις δύο στάσεις μέσω της ίδιας διαδρομής, κάτι που είναι αρκετά συνηθισμένο σε λεωφορειακές γραμμές. Το συγκεκριμένο πρόβλημα λοιπόν προκύπτει διότι στα λεωφορεία και τα τρόλεϊ έχει δοθεί μία ενιαία μέση ταχύτητα, συνεπώς ο

αλγόριθμος αντιλαμβάνεται τις ακμές ως ίσου κόστους και δεν λαμβάνει υπ' όψιν το κόστος μετεπιβίβασης. Αλλά και κατόπιν της εφαρμογής του αλγορίθμου δεν μπορεί να γίνει ο σωστός υπολογισμός αφού δεν είναι γνωστό ποιες είναι ορθές και ποιές άσκοπες ή τυχαίες μετεπιβιβάσεις.

Συνεπώς πρέπει να δομηθεί ένα σύστημα διόρθωσης των αποτελεσμάτων του Dijkstra και υπολογισμού του κόστους κάθε κόμβου.

Λογισμικό και Προγράμματα που χρησιμοποιήθηκαν

Όπως αναφέρθηκε στο προηγούμενο εδάφιο πρέπει να δομηθεί ένας αλγόριθμος ο οποίος θα δέχεται τα αποτελέσματα από την εκτέλεση του Dijkstra, θα τα ελέγχει και θα τα διορθώνει όπου χρειάζεται. Επιπλέον για την εξοικονόμηση χρόνου, θα πρέπει ο Dijkstra να καλείται αυτόματα για κάθε ζεύγος κόμβων του δικτύου, αφού οριστεί ο κόμβος αφετηρίας.

Με δεδομένο ότι ο όγκος των πληροφοριών είναι αρκετά μεγάλος και η επεξεργασία τους σημαντικά πολύπλοκη επιλέχθηκε να δομηθεί ο αλγόριθμος στη γλώσσα προγραμματισμού Python.

Νέος αλγόριθμος προσδιορισμού χρόνου βέλτιστης διαδρομής

Ο νέος αλγόριθμος που δομήθηκε ακολουθεί την εξής μεθοδολογία. Αρχικά εφαρμόζεται ο αλγόριθμος Dijkstra από τον κόμβο αφετηρίας προς όλους τους υπόλοιπους κόμβους του δικτύου. Αυτό καθίσταται δυνατό με μία δομή επανάληψης, που διαβάζει τους κόμβους του γράφου έναν έναν, ορίζει κάθε φορά τον τρέχοντα κόμβο ως κόμβο προορισμού και καλεί τον Dijkstra. Ύστερα προχωράει στον επόμενο κόμβο και επαναλαμβάνεται η διαδικασία.

Οι διαδρομές που δημιουργούνται εξάγονται σε ένα συνολικό αρχείο. Η κάθε διαδρομή αποτελεί μία λίστα που περιλαμβάνει την αλληλουχία των ακμών που διασχίστηκαν, το κωδικό της γραμμής που χρησιμοποιήθηκε, τον κόμβο από τον οποίο ξεκινάει η ακμή (source) και τον κόμβο στον οποίο καταλήγει (target).

Σε αυτό το σημείο εισέρχεται η κύρια παρέμβαση του προγράμματος που δομήθηκε. Για κάθε δημιουργημένη διαδρομή εκτελείται ο έλεγχος των μετεπιβιβάσεων. Ουσιαστικά ελέγχεται η λίστα της διαδρομής σε σχέση με τη λίστα του γράφου. Δηλαδή όπου το σύστημα εντοπίσει αλλαγή στον κωδικό της γραμμής ελέγχει αν στη λίστα του γράφου υπάρχει η συγκεκριμένη γεωμετρία ακμής με κωδικό γραμμής ίδιον με της προηγούμενης. Αν ο έλεγχος επιστρέψει TRUE ο κωδικός γραμμής αλλάζει και ακυρώνεται η μετεπιβίβαση που είχε εισαχθεί τυχαία από τον Dijkstra. Είναι ευνόητο ότι αν ο έλεγχος επιστρέψει FALSE το πρόγραμμα δεν προβαίνει σε καμία αλλαγή και προχωρά στον έλεγχο της επόμενης ακμής. Προκειμένου να αυξηθεί η αποτελεσματικότητα του νέου αλγορίθμου, δομήθηκε αναδρομικά. Δηλαδή όταν τελειώνει ο έλεγχος της λίστας, ο αλγόριθμος καλεί τον εαυτό του και ξανατρέχει για την επόμενη λίστα. Αυτό είναι κάτι ιδιαίτερα σημαντικό καθώς δεν θα ήταν δυνατό, αλλά και ούτε αποδοτικό να καλεστεί ο αλγόριθμος χειροκίνητα χιλιάδες φορές, όσες είναι οι παραχθείσες διαδρομές.

Στο προηγούμενο στάδιο παραλείφθηκε ένα καίριο βήμα του αλγορίθμου. Όταν τερματιστεί ο έλεγχος και η διόρθωση μίας διαδρομής και πριν το σύστημα προχωρήσει στην επόμενη, υπολογίζεται το συνολικό κόστος της και αποδίδεται στον κόμβο

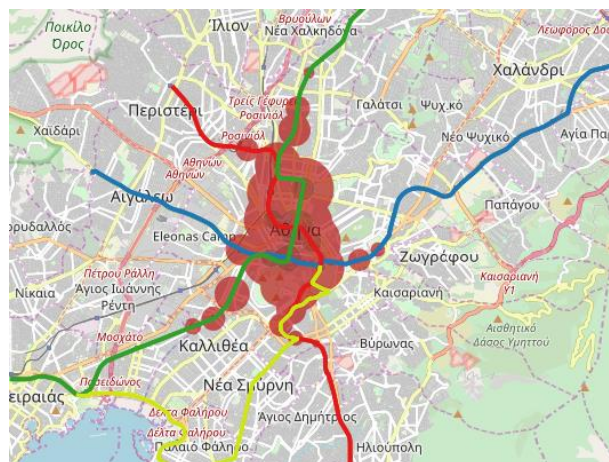
προορισμού. Η διαδικασία αυτή εκτελείται με μία ακόμη δομή επανάληψης. Διαβάζονται οι κωδικοί των γραμμών και προστίθενται διαδοχικά τα κόστη διάσχισης των ακμών. Επιπλέον κάθε φορά που εντοπίζεται μετεπιβίβαση προστίθεται και αυτό το επιπρόσθετο κόστος στο συνολικό. Ο χρόνος μετεπιβίβασης έχει ορισθεί ως ο μισός της συχνότητας κάθε γραμμής. Τελικά σε ένα νέο αρχείο γράφεται ο κωδικός του τελευταίου κόμβου της διαδρομής (κόμβος προορισμού) και δίπλα το συνολικό του κόστος. Το αρχείο αυτό εξάγεται σε txt και είναι έτοιμο προς περαιτέρω επεξεργασία στη βάση δεδομένων και τα προγράμματα γεωγραφικών συστημάτων πληροφοριών.

Υπολογισμός χρονικού κόστους κόμβων

Αφού έχουν διορθωθεί οι διαδρομές και δεν υπάρχουν άσκοπες μετεπιβιβάσεις, πρέπει να υπολογιστεί το τελικό κόστος κάθε κόμβου. Όταν καλείται από το πρόγραμμα η αντίστοιχη συνάρτηση σαρώνονται όλες οι εγγραφές της λίστας της διαδρομής και προστίθενται διαδοχικά τα κόστη διάσχισης των ακμών. Παράλληλα διαβάζεται ο κωδικός κάθε γραμμής και όταν υπάρχει αλλαγή προστίθεται στο κόστος ο χρόνος μετεπιβίβασης. Συμβατικά έχει θεωρηθεί ότι ο χρόνος αυτός είναι ο μισός από την τυπική συχνότητα της γραμμής.

Σχεδιασμός Ισοχρονικών καμπυλών

Το πιο σημαντικό βήμα στην εκτέλεση της εφαρμογής είναι, με δεδομένο το χρονικό κόστος των κόμβων, ο σχεδιασμός των ισοχρονικών καμπυλών που είναι και ο σκοπός της εργασίας. Για το σκοπό αυτό δοκιμάστηκαν κάποιοι αλγόριθμοι και μέθοδοι ώστε να επιλεγεί εκείνος που μπορεί να δώσει πιο ακριβή και ευανάγνωστα αποτελέσματα. Συγκεκριμένα δοκιμάστηκαν οι αλγόριθμοι Convex Hull, Concave Hull και η μέθοδος δημιουργίας ζωνών (buffers), η οποία τελικά και επιλέχθηκε, λόγω ποικίλων πλεονεκτημάτων που παρουσιάζει.

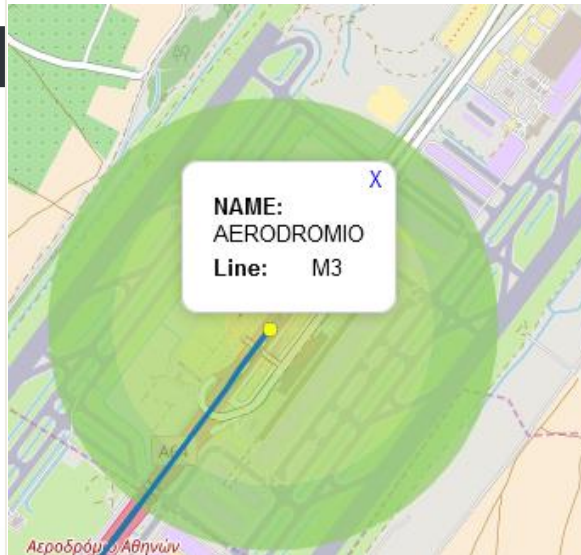
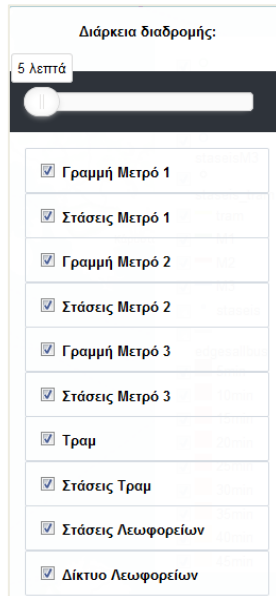


Εικόνα 2 : Παρουσίαση ισοχρονικών καμπυλών μέσω δημιουργίας ζωνών

Δημιουργία εφαρμογής

Για την παρουσίαση των αποτελεσμάτων της ανάλυσης δημιουργήθηκε μία ιστοσελίδα στην οποία παρέχεται η δυνατότητα διάδρασης με το χρήστη. Μέσω των HTML και CSS μορφοποιήθηκε η εφαρμογή ενώ μέσω της JavaScript, ορίστηκαν οι λειτουργίες της. Πιο

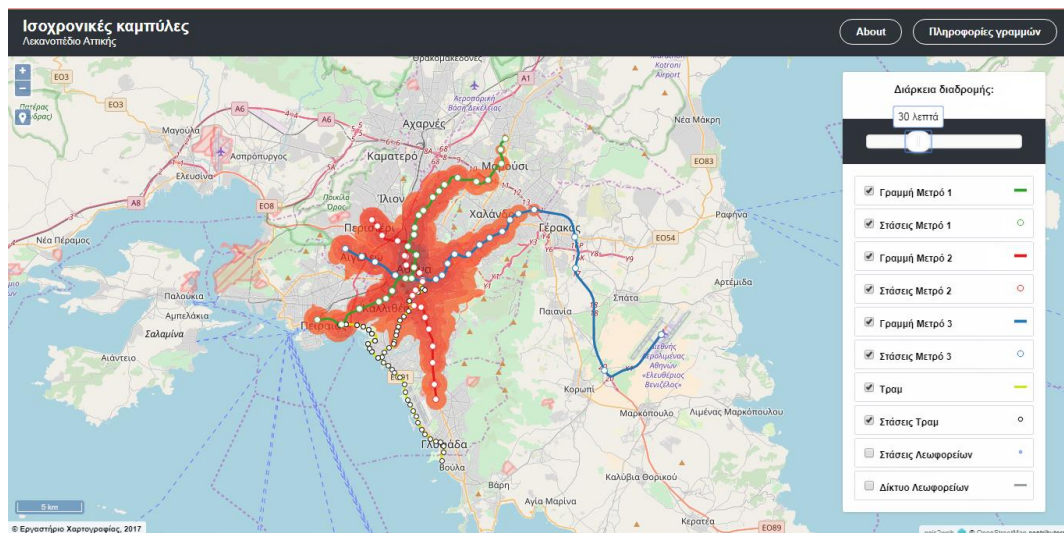
συγκεκριμένα ο χρήστης έχει τη δυνατότητα, μέσω ενός κέρσορα (Εικόνα 3), να ορίσει το χρόνο διαδρομής που επιθυμεί και θα εμφανιστεί στον χάρτη του OSM η αντίστοιχη ισochρονική καμπύλη. Επίσης έχει εισαχθεί και η πληροφορία των γραμμών μετακίνησης και των στάσεων του δικτύου της Αθήνας. Ο χρήστης μπορεί να επιλέξει μέσω μιας λίστας (Εικόνα 3) τι από αυτά επιθυμεί να είναι ορατό στον χάρτη και κινώντας τον κέρσορα πάνω σε κάποιο στοιχείο ενός θεματικού επιπέδου, αυτό θα επισημανθεί και παράλληλα θα εμφανιστεί ένα παράθυρο με τις βασικές πληροφορίες του στοιχείου αυτού (Εικόνα 4).



Εικόνα 3

Εικόνα 4

Η τελική μορφή της ιστοσελίδας φαίνεται στην Εικόνα 5.



Εικόνα 5 : Ορισμός του χρόνου μετακίνησης στα 30 λεπτά και εμφάνιση των αντίστοιχων ισochρονικών καμπυλών

Η ιστοσελίδα της εφαρμογής είναι η :

http://atlas.geocenter.survey.ntua.gr:8081/oasa_timemap/index.html

Βιβλιογραφικές αναφορές

V. Bauer, J. Gamper, R. Loperfido, S. Profanter, S. Putzer, I. Timko, Computing isochrones in multi-modal, schedule-based transport networks, GIS '08, Nov. 2008

Valentin Buchhold (B), Julian Dibbelt, and Dorothea Wagner Karlsruhe Fast Exact Computation of Isochrones in Road Networks Moritz Baum, Institute of Technology, Karlsruhe, Germany
{moritz.baum,valentin.buchhold,julian.dibbelt,dorothea.wagner}@kit.edu

Aldous M. Joan, Wilson J. Robin (2000), Graphs and Applications: An Introductory Approach, Springer Publications, London.

Jin-Seo Park and Se-Jong Oh, A New Concave Hull Algorithm and Concaveness Measure for n-dimensional Datasets, Journal of Information Science and Engineering, Vol. 28 No. 3, pp. 587-600, May 2012

O'Reilly Radar , Programming Language Trends, Radar.oreilly.com. 2 August 2006

Biggs, N.; Lloyd, E. and Wilson, R. (1986). Graph Theory, 1736-1936. Oxford University Press.

Cormen H. Thomas, Leiserson E. Charles, Rivest L. Ronald and Stein Clifford (2001), Introduction to Algorithms, The MIT Press, Massachusetts Institute of Technology.

Reinhard Diestel, Graph Theory, Springer GTM 173, 5th edition 2016

L'Huillier, S.-A.-J. (1861). «Mémoire sur la polyèdrométrie». Annales de Mathématiques 3: 169–189.

Johann Gamper et al., De_ning isochrones in multimodal spatial networks, Proceedings of the 20th ACM international conference on Information and knowledge management, CIKM 2011, Glasgow, Scotland

Veronika Bauer et al., Computing isochrones in multi-modal, schedule-based transport networks, Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems. GIS '08. Irvine, California: ACM, 2008,

Johann Gamper, Michael Bohlen, and Markus Innerebner, Scalable Computation of Isochrones with Network Expiration, Proceedings of the 24th International Conference on Scientific and Statistical Database Management. SSDBM' 12. Chania, Crete, Greece: Springer-Verlag, 2012,.

Σχεδιασμός αλγορίθμων, John Kleinberg, Eva Tardos, 2008

Στεφανάκης Εμμανουήλ, Βάσεις Γεωγραφικών Δεδομένων και Συστήματα Γεωγραφικών Πληροφοριών, Εκδόσεις Παπασωτηρίου, Αθήνα, 2013.

Κολιός Ν. (2009), Χωρική Βάση Δεδομένων PostgreSQL/ PostGIS και Σύστημα Γεωγραφικών Πληροφοριών Quantum GIS, Οδηγός Χρήσης, Εταιρεία Ελεύθερου Λογισμικού/ Λογισμικού Ανοικτού Κώδικα- Creative Commons Attribution- Non Commercial- ShareAlike 3.0- Ελλάδα

Στεφανάκης Εμμανουήλ, Τεχνολογίες Δημοσιοποίησης Χαρτογραφικού Περιεχομένου στον Παγκόσμιο Ιστό, Εκδόσεις Νέων Τεχνολογιών, 2009

Shekhar Shashi, Chawla Sanjay (2003), Spatial Databases: A Tour, Pearson Education Inc., Upper Saddle River, New Jersey 07458.

openlayer's 2.10 Beginner's Guide

pgRouting, Practical Guide, 2016

Musciano, C., and Kennedy, B., 2006. HTML & XHTML: The Definitive Guide. O'Reilly Media, Inc.; 6 edition

Marijin Haverbeke, Eloquent JavaScriptsecond edition, 2011

Ιστοσελίδες

Algorithms and Data Structures:

http://www.algolist.net/Data_structures/Graph

Google Maps:

<http://www.googlemaps.com>

Graph Theory Springer GTM 173, 5th edition 2016

<http://diestel-graph-theory.com/basic.html?>

Mapumental :

<http://www.mapumental.com>

Wikipedia:

https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

<https://el.wikipedia.org/wiki/PostgreSQL>

https://en.wikipedia.org/wiki/Bellman%E2%80%93Ford_algorithm

Oasa official site :

http://www.ametro.gr/?page_id=10

Oasa telematics site :

<http://telematics.oasa.gr/>

pgRouting Official Site :

<http://pgrouting.org/>

PostgreSQL official site :

<https://www.postgresql.org/>

PostGIS Manual 1.3.6:

<http://postgis.refractor.net/documentation/manual-1.3/>

MySQL official site :

<https://www.mysql.com/>

pgAdmin official site :

<https://www.pgadmin.org/>

Documentation for QGIS 2.18 :

<http://docs.qgis.org/2.18/en/docs/>

Python for Beginners :

<https://www.python.org/about/gettingstarted/>

Python Tutorials :

https://www.tutorialspoint.com/python/python_basic_syntax.htm

w3schools :

<https://www.w3schools.com/>

HTML Language Specification:

<http://www.w3.org/html/>

Css Specification:

<http://www.wlearn.gr/index.php/home-css-83>

JavaScript official site:

<https://www.javascript.com/>

Regular expressions of JavaScript :

<https://regexr.com/>

OpenLayers official site :

<https://openlayers.org/>

JavaScript Range Slider :

<https://refreshless.com/nouislider/>

GitHub, [The world's leading software development platform](https://github.com/econchick/4666413) :

<https://gist.github.com/econchick/4666413>

<https://github.com/iotacss/iotacss>

Algorithm creation and cartographic map rendering of the time travel by public transport in Attica

Abstract

The main objective of this study is to create maps that show curves of equal travel times through the Public Transit Network of Athens starting from Omonoia Square. We aim to produce a realistic analysis of the results based on a reasonable estimate of travel times.

An isochrone is defined as a polygon on the map that includes all the areas that can be accessed within a certain amount of time from an initial point. The processes and the parameters of isochrones' creation depend on the network based on which the analysis is conducted. Most published studies have applied this specific analysis on road networks.

As part of this study an original algorithm was developed for the calculation of isochrones in multimodal networks. This algorithm is based on Dijkstra's routing algorithm. The specific characteristics of the studied network such as multiple edges (parts of bus routes) of the same time cost that may be located between two nodes (stations), the additional costs involved in cases of changing the type of transit line, the fact that the stations (nodes) do not lie on the network (edges) as it should be for the analysis through shortest path algorithms

and other considerations that emerged during the procedure have been indexed and addressed.

Additionally, an online cartographic application was designed and implemented to illustrate the created isochrones. The user interface will allow the user to have the option of selecting the desired travel time. The user will also have the option to enable multiple layers on the map which will display the map of the corresponding mode of transport.