



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΝΕΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ»**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Εύρεση βέλτιστων δρομολογίων σε σύστημα αστικών
συγκοινωνιών με βάση το πρότυπο GTFS**

Σωκράτης Ε. Αριστοβουλίδης

Επιβλέπων: Αλέξης Δελής, Καθηγητής

ΑΘΗΝΑ

ΟΚΤΩΒΡΗΣ 2011

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Εύρεση βέλτιστων δρομολογίων σε σύστημα αστικών συγκοινωνιών με βάση το πρότυπο GTFS

Σωκράτης Ε. Αριστοβουλίδης

A.M.: M932

ΕΠΙΒΛΕΠΩΝ: **Αλέξης Δελής, Καθηγητής**

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: **Αλέξης Δελής, Καθηγητής**
Παναγιώτης Ροντογιάννης, Αναπληρωτής καθηγητής

ΑΘΗΝΑ

ΟΚΤΩΒΡΗΣ 2011

ΠΕΡΙΛΗΨΗ

Ενώ το πρόβλημα της εύρεσης της συντομότερης χρονικά διαδρομής σε ένα σύστημα συγκοινωνιών είναι έως ένα βαθμό συναφές με το πρόβλημα εύρεσης της βέλτιστης διαδρομής σε ένα οδικό δίκτυο παρόλα αυτά δεν αντιμετωπίζεται από τους κλασσικούς αλγόριθμους που έχουν αναπτυχθεί για το σκοπό αυτό. Σημαντικός παράγοντας διαφοροποίησης των δύο προβλημάτων είναι η χρονική διάσταση που εισάγεται από το πρώτο πρόβλημα.

Ένα οδικό δίκτυο μπορεί να παρασταθεί με έναν κατευθυνόμενο γράφο και η βέλτιστη διαδρομή μεταξύ δύο σημείων μπορεί να βρεθεί με τη χρήση ενός κλασσικού αλγόριθμου (πχ Dijkstra, A*). Ένα σύστημα συγκοινωνιών εξυπηρετείται από ένα σύνολο γραμμών κάθε μία από τις οποίες απαρτίζεται από ένα σύνολο δρομολογίων τα οποία μπορούν να μεταβάλλονται ανάλογα με την ημέρα και την ώρα. Το σύστημα περιγράφεται από το σύνολο όλων των στάσεων που το απαρτίζουν, τις γραμμές και το πρόγραμμα δρομολογίων κάθε γραμμής. Η εύρεση της βέλτιστης διαδρομής δεν εξαρτάται μόνο από την τοπολογία του δικτύου που σχηματίζεται από τις στάσεις του συστήματος, αλλά και από τη χρονική συνάφεια των διελεύσεων των συγκοινωνιακών μέσων από αυτές.

Η παρούσα εργασία προτείνει έναν αλγόριθμο εύρεσης της βέλτιστης διαδρομής (και τυχόν εναλλακτικών διαδρομών με παρεμφερές κόστος) σε ένα σύστημα συγκοινωνιών. Σε αντίθεση με τους κλασσικούς αλγόριθμους επίλυσης γράφων οι οποίοι εξετάζουν διαδοχικούς κόμβους, η αναζήτηση της βέλτιστης διαδρομής γίνεται με την εκτίμηση του κόστους των διαδρομών που σχηματίζονται από διαδοχικά (χωρικά και χρονικά) υποσύνολα δρομολογίων όπως αυτά αποτυπώνονται στο πρόγραμμα δρομολογίων του συστήματος.

Στο πλαίσιο της εργασίας έχει γίνει υλοποίηση του αλγόριθμου ο οποίος μπορεί να εκτελεστεί μέσω ενός web interface και να παρέχει τη βέλτιστη διαδρομή μεταξύ δύο σημείων και για δεδομένη ημέρα και ώρα αναχώρησης για το San Francisco και την Αθήνα.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Αλγόριθμοι εύρεσης δρομολογίων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: σύστημα συγκοινωνιών, σχεδιασμός διαδρομής, Γεωγραφικό Σύστημα Πληροφοριών, δρομολόγιο, αλγόριθμος

ABSTRACT

Though the problem of finding the best (quickest) path in an urban transit system is related to some degree to the problem of finding the best route in a road network, the first problem is not addressed adequately from the classic graph solving algorithms that are used to solve the second problem. An important difference between these two problems is the temporal dimension introduced by the first problem.

A road network can be represented by a directed graph and the optimal route between two points can be found using a classical algorithm (Dijkstra, A *). An urban transit system is served by a set of lines each of which consists of a set of services that can be changed depending on the day and time according to the transit system schedule. The transit system is described by the set of stops, routes, trips and the timetable of each trip. Finding the optimal path depends not only on the topology of the network formed by the stops of the system, but also on the temporal relevance of the means of transport crossing them.

This essay proposes an algorithm for finding the best route (and any alternative paths with similar costs) in a transit system. In contrast to the classical graph-solving algorithms which examine subsequent nodes of the graph, the search for the optimal path is performed estimating the cost of the paths formed by successive (spatially and temporally) subsets services as reflected in the schedule of the trips of the transit system.

As part of this work, the proposed algorithm has been implemented and can be run from a web interface finding the best transit schedule between two locations for a given time of departure for the cities of San Francisco and Athens.

SUBJECT AREA: Trip planning algorithms

KEYWORDS: transit system, trip planning, GIS, route, algorithm

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ.....	7
2. ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ	9
2.1 Το πρότυπο General Transit Feed Specification (GTFS).....	9
2.1.1 Γενικά	9
2.1.2 Το σχήμα δεδομένων του προτύπου GTFS.....	9
2.1.3 Επέκταση του σχήματος GTFS.....	11
2.2 Απόσταση μεταξύ δύο σημείων.....	11
3. ΑΛΓΟΡΙΘΜΟΣ	12
3.1 Δεδομένα εισόδου	12
3.2 Διαδρομή.....	12
3.3 Ευρετικές Μέθοδοι (Heuristics).....	12
3.3.1 Κανόνας απόστασης	12
3.3.2 Λοιποί περιορισμοί.....	13
3.4 Υπολογισμός χρόνου βαδίσματος	13
3.5 Βαθμολόγηση διαδρομής	13
3.6 Περιγραφή εκτέλεσης του αλγορίθμου.....	13
3.6.1 Αρχικοποίηση της μνήμης	13
3.6.2 Εκτέλεση του αλγόριθμου	14
3.6.3 Σχηματική περιγραφή του αλγόριθμου	16
4. ΥΛΟΠΟΙΗΣΗ	18
4.1 Αρχιτεκτονική.....	20
4.2 Παραδείγματα.....	21
4.3 Εργαλεία υλοποίησης.....	23
5. ΜΕΛΛΟΝΤΙΚΕΣ ΒΕΛΤΙΩΣΕΙΣ	24
5.1 Σμίκρυνση βάσης δεδομένων GTFS	24
5.2 Προτίμηση μέσω σταθερής τροχιάς.....	24
5.3 Δυναμικές ευρετικές μέθοδοι	24
ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ.....	26
ΑΝΑΦΟΡΕΣ	27

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Σχήμα δεδομένων του αλγορίθμου.	9
Εικόνα 2: Εκτέλεση αλγόριθμου - 1ο βήμα.	16
Εικόνα 3: Εκτέλεση αλγόριθμου - 2ο βήμα.	17
Εικόνα 4: Εκτέλεση αλγόριθμου - 3ο βήμα.	17
Εικόνα 5: Εκτέλεση αλγόριθμου - 4ο βήμα.	18
Εικόνα 6: Εκτέλεση αλγόριθμου - 4ο βήμα (επανάληψη).....	18
Εικόνα 7: Αποτέλεσμα αλγόριθμου.....	19
Εικόνα 8: Επεξεργασία δεδομένων GTFS.	20
Εικόνα 9: Αρχιτεκτονική της εφαρμογής.	21
Εικόνα 10: Αναζήτηση δρομολογίου στην Αθήνα.	22
Εικόνα 11: Αναζήτηση δρομολογίου στο San Francisco.....	22

1. ΕΙΣΑΓΩΓΗ

Καθώς τα αστικά κέντρα γίνονται ολοένα μεγαλύτερα σε έκταση και σε πυκνότητα, απαιτούν πολυπλοκότερα συστήματα συγκοινωνιών, η δρομολόγηση εντός των οποίων δεν είναι πάντα απλή υπόθεση ιδίως για κάποιον ο οποίος δεν έχει το χρόνο να εξοικειωθεί με τα δρομολόγια ή / και χρειάζεται να χρησιμοποιήσει το σύστημα για να μεταβεί σε πολλά διαφορετικά σημεία σε διάστημα λίγων ημερών, όπως πχ ένας επισκέπτης μιας πόλης.

Μια από τις δημοφιλέστερες εφαρμογές των συστημάτων γεωγραφικών πληροφοριών (GIS) είναι η εύρεση της συντομότερης διαδρομής μεταξύ δύο σημείων διαμέσου ενός οδικού δικτύου. Το πρόβλημα αυτό μπορεί να αναχθεί στο μαθηματικό πρόβλημα της εύρεσης της βέλτιστης διαδρομής μεταξύ δύο κόμβων ενός γράφου και επιλύεται από μια πλειάδα αλγορίθμων [1], [2], [3] οι οποίοι έχουν αναπτυχθεί σε βάθος χρόνου αρκετών δεκαετιών.

Μια ιδιαίτερα χρήσιμη εφαρμογή ενός συστήματος GIS είναι η εύρεση της βέλτιστης διαδρομής σε ένα σύστημα συγκοινωνιών με δεδομένα τα σημεία αναχώρησης και προορισμού καθώς, την ώρα αναχώρησης καθώς και το πρόγραμμα δρομολογίων του συστήματος συγκοινωνιών. Το πρόβλημα αυτό παρουσιάζει ορισμένες ομοιότητες με αυτό της δρομολόγησης σε ένα οδικό δίκτυο, αλλά έχει και ορισμένες σημαντικές διαφορές οι οποίες δεν επιτρέπουν την αποτελεσματική επίλυσή του με τη χρήση των κλασσικών αλγορίθμων επίλυσης γράφου που αναφέρθηκαν παραπάνω. Οι λόγοι που κάτι τέτοιο δεν είναι εφικτό όπως αυτοί καταγράφονται στο [4] είναι οι εξής:

- **Τοπολογία:** Ένα σύστημα συγκοινωνιών δεν μπορεί να μοντελοποιηθεί αποτελεσματικά με τη χρήση γράφου ώστε να χρησιμοποιηθεί από τους κλασσικούς αλγόριθμους επίλυσης λόγω του ότι μεταξύ δύο κόμβων (στάσεων) υπάρχουν περισσότερες από μία συνδέσεις.
- **Δυναμική:** Ένα σύστημα συγκοινωνιών μεταβάλλεται στη διάρκεια της ημέρας σύμφωνα με το πρόγραμμα δρομολογίων, σε αντίθεση με ένα οδικό δίκτυο.
- **Ορισμός βέλτιστου μονοπατιού:** Ακόμη και αν το σύστημα μπορούσε να μοντελοποιηθεί τοπολογικά με τη χρήση γράφου, ο χρόνος αναμονής ή μετάβασης σε άλλη γραμμή δεν μπορεί να μοντελοποιηθεί. Η διαδρομή με το μικρότερο χρόνο ταξιδιού δεν είναι απαραίτητα και η βέλτιστη.

Ενώ υπάρχει εκτενής βιβλιογραφία για το πρόβλημα της επίλυσης γράφου, υπάρχει σχετικά περιορισμένη βιβλιογραφία, πχ [5], [6], [4], [7], [8] για το δεύτερο πρόβλημα. Στην παρούσα εργασία προτείνεται ένας αλγόριθμος εύρεσης της βέλτιστης διαδρομής (και τυχόν εναλλακτικών διαδρομών με παρεμφερές κόστος) σε ένα σύστημα συγκοινωνιών. Σε αντίθεση με τους κλασσικούς αλγόριθμους επίλυσης γράφων οι οποίοι εξετάζουν διαδοχικούς κόμβους, η αναζήτηση της βέλτιστης διαδρομής γίνεται με την εκτίμηση του κόστους των διαδρομών που σχηματίζονται από διαδοχικά (χωρικά

και χρονικά) υποσύνολα δρομολογίων όπως αυτά αποτυπώνονται στο πρόγραμμα δρομολογίων του συστήματος συγκοινωνιών.

Ένα σύστημα συγκοινωνιών απαρτίζεται από τα εξής συστατικά:

- **Τις στάσεις:** Τα γεωγραφικά σημεία τα οποία επισκέπτονται τα μέσα του συστήματος
- **Τα δρομολόγια:** Πρόκειται για διακριτές ακολουθίες επισκέψεων που πραγματοποιεί ένα συγκεκριμένο μέσο του συστήματος σε ένα υποσύνολο των στάσεων.
- **Τις γραμμές:** Απαρτίζονται από ένα ή περισσότερα δρομολόγια.
- **Το πρόγραμμα:** Καθορίζει το χρόνο πραγματοποίησης των δρομολογίων.

Η λύση του προβλήματος της εύρεσης της βέλτιστης διαδρομής στο σύστημα είναι μια ακολουθία από υποσύνολα δρομολογίων που πραγματοποιούνται σε συγκεκριμένο χρόνο έτσι ώστε η μετάβαση από το σημείο αναχώρησης στο σημείο προορισμού να διαρκέσει το λιγότερο δυνατό.

Η κεντρική ιδέα του αλγόριθμου που παρουσιάζεται εδώ, έχει ομοιότητες με την εργασία του R. Huang [4] ως προς το ότι η αναζήτηση γίνεται όχι με την επίσκεψη των διαδοχικών κόμβων ενός γράφου, αλλά με την αξιολόγηση διαδοχικών δρομολογίων του συστήματος. Ο αλγόριθμος που παρουσιάζεται εδώ, χρησιμοποιεί ως μοντέλο δεδομένων το GTFS, το οποίο είναι ευρέως διαδεδομένο με δεκάδες συγκοινωνιακούς οργανισμούς ανά τον κόσμο να έχουν διαθέσει τα δεδομένα των δρομολογίων τους σε αυτό. Επίσης τα δρομολόγια που είναι ενεργά δεν ανακτώνται εξ αρχής με την υπόθεση ότι η διαδρομή θα διαρκέσει συγκεκριμένο χρονικό διάστημα, αλλά με βάση την ώρα που ο επιβάτης βρίσκεται στο σημείο εκκίνησης ή μετάβασης μεταξύ δύο δρομολογίων, με κατάλληλα ερωτήματα στη βάση δεδομένων που χρησιμοποιεί ο αλγόριθμος.

2. ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

2.1 Το πρότυπο General Transit Feed Specification (GTFS)

2.1.1 Γενικά

Βασικός παράγοντας του προβλήματος της εύρεσης της βέλτιστης διαδρομής είναι η ύπαρξη λεπτομερών χρονικών και γεωγραφικών δεδομένων όλων των γραμμών και δρομολογίων του συστήματος. Εξίσου σημαντικό είναι τα δεδομένα αυτά να έχουν μία ενιαία και εύκολα επεξεργάσιμη μορφή.

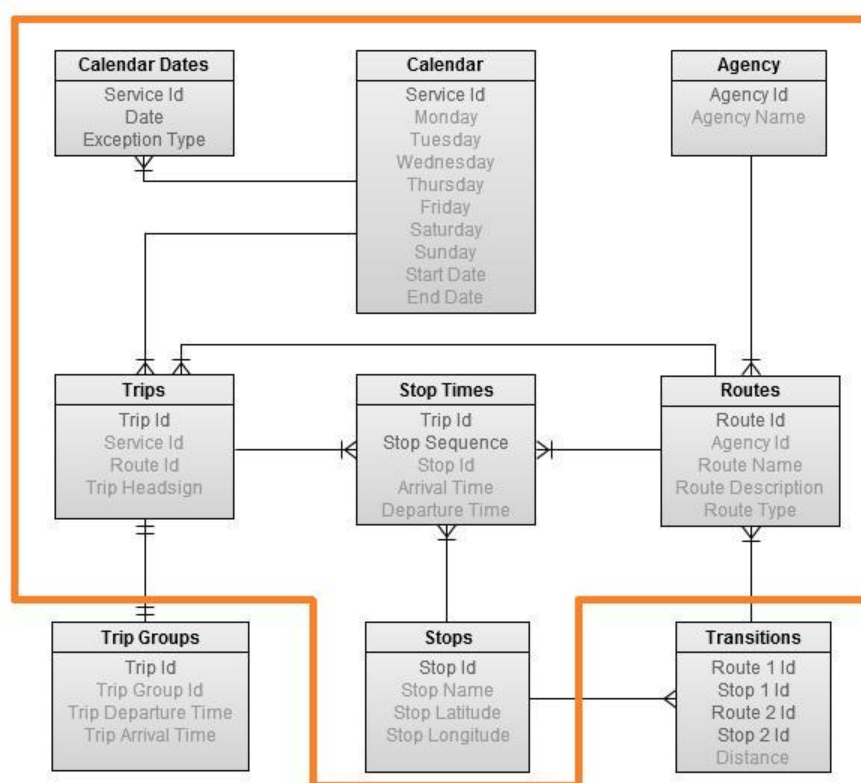
Για την επεξεργασία των δεδομένων χρησιμοποιήθηκε το πρότυπο General Transit Feed Specification (GTFS) [9]. Επιπρόσθετα έγινε μια μικρή επέκταση του σχήματος δεδομένων του προτύπου για λόγους απόδοσης του αλγορίθμου. Παρακάτω θα περιγραφεί αναλυτικά το GTFS.

Για το πρότυπο αυτό υπάρχουν διαθέσιμα δεδομένα για μερικές δεκάδες πόλεις τα οποία ανανεώνονται σε τακτά χρονικά διαστήματα από τους ενδιαφερόμενους φορείς συγκοινωνιών. Για την υλοποίηση και δοκιμή του αλγορίθμου χρησιμοποιήθηκαν δεδομένα για το σύστημα συγκοινωνιών του San Francisco [10].

2.1.2 Το σχήμα δεδομένων του προτύπου GTFS.

Στο παρακάτω διάγραμμα απεικονίζονται οι πίνακες που αποτελούν το σχήμα δεδομένων του προτύπου καθώς και οι δύο πίνακες που αποτελούν την επέκταση που έχει γίνει.

GTFS Entity Relationship Diagram



Εικόνα 1: Σχήμα δεδομένων του αλγορίθμου

Οι πίνακες *Trip Groups* και *Transitions* αποτελούν προσθήκη στο σχήμα δεδομένων του προτύπου.

Παρακάτω ακολουθεί περιγραφή των πινάκων που χρησιμοποιεί το πρότυπο GTFS.

Agency

Περιέχει στοιχεία για τους οργανισμούς συγκοινωνιών που εξυπηρετούν το σύστημα. Πχ ΑΤΤΙΚΟ ΜΕΤΡΟ ή ΕΘΕΛ.

Routes

Περιέχει τις γραμμές που εξυπηρετούν το σύστημα αντιστοιχίζοντάς τις με τον οργανισμό στον οποίο ανήκουν. Πχ. γραμμή 2 του μετρό η οποία αντιστοιχίζεται στην ΑΤΤΙΚΟ ΜΕΤΡΟ.

Trips

Περιέχει όλα τα δρομολόγια που εκτελούνται από τις γραμμές του συστήματος. Ως δρομολόγιο θεωρούμε μια ακολουθία στάσεων που εκτελεί ένα μέσο (πχ τραίνο) μιας γραμμής **κάθε μία από τις οποίες πραγματοποιείται σε συγκεκριμένο χρόνο**. Μια γραμμή (route) πραγματοποιεί σε μία ημέρα πολλά δρομολόγια τα οποία μπορούν να αποτελούνται από διαφορετικές ακολουθίες στάσεων. Πχ η γραμμή 2 του ΜΕΤΡΟ εκτελεί το δρομολόγιο Πειραιάς – Κηφισιά 35 φορές και άλλες τόσες το δρομολόγιο Κηφισιά Πειραιάς. Εκτελεί επίσης από 10 φορές το δρομολόγιο Πειραιάς – Ειρήνη και 6 φορές το δρομολόγιο Ειρήνη – Πειραιάς. Ο πίνακας trips θα περιείχε $35 \cdot 2 + 10 + 6 = 86$ εγγραφές για τα παραπάνω δρομολόγια τα οποία θα αντιστοιχίζονταν με το route Id της γραμμής 2 και το καθένα θα είχε μοναδικό κλειδί το πεδίο trip Id.

Stops

Περιέχει τις όλες τις στάσεις του συστήματος. Για κάθε στάση ορίζεται μοναδικό κλειδί (stop Id), όνομα και γεωγραφικές συντεταγμένες.

Stop Times

Περιέχει την ακριβή χρονική πληροφορία για όλες τις στάσεις όλων των δρομολογίων που εκτελούνται από τις γραμμές του συστήματος.

Calendar

Κάθε δρομολόγιο (trip) συσχετίζεται με έναν κωδικό υπηρεσίας. Σε αυτό τον πίνακα ορίζεται για κάθε ημέρα της εβδομάδας αν η υπηρεσία είναι ενεργή ή όχι. Αν για παράδειγμα θέλουμε για όλο το σύστημα να έχουμε τρία διαφορετικά σύνολα δρομολογίων τις καθημερινές, το Σάββατο και την Κυριακή, ορίζουμε τρεις κωδικούς υπηρεσίας (service Id) και ποιες ημέρες είναι ενεργός ο καθένας. Στη συνέχεια συσχετίζουμε σύνολα δρομολογίων με τους κωδικούς υπηρεσίας που ορίσαμε.

Calendar Dates

Σε αυτόν τον πίνακα ορίζονται τυχόν εξαιρέσεις από τους κανόνες που αποτυπώνονται στον πίνακα Calendar. Πχ αν μία επίσημη αργία είναι μέσα στην εβδομάδα αλλά τα δρομολόγια θέλουμε να είναι σαν της Κυριακής τότε ορίζουμε κατάλληλα την εξαίρεση στον πίνακα Calendar Dates.

2.1.3 Επέκταση του σχήματος GTFS

Όπως αναφέρθηκε και νωρίτερα, το σχήμα δεδομένων του προτύπου GTFS επεκτάθηκε κατάλληλα για τις ανάγκες του αλγορίθμου. Η επέκταση αυτή αποτελείται από τους τρεις παρακάτω πίνακες. Τα δεδομένα των πινάκων αυτών παράγονται από την επεξεργασία των βασικών πινάκων του σχήματος GTFS.

Route Has Stops

Περιέχει όλες τις στάσεις κάθε διαδρομής. Προκύπτει από τη συσχέτιση των πινάκων Routes, Trips και Stop Times.

Trip Groups

Επειδή τα δρομολόγια μεταβάλλονται σύμφωνα με την ημέρα και την ώρα, ομαδοποιήθηκαν με βάση την γραμμή στην οποία ανήκουν και την ακολουθία των στάσεων από τις οποίες αποτελούνται. Δύο δρομολόγια ανήκουν στην ίδια ομάδα (trip group) αν ανήκουν στην ίδια γραμμή και αν αποτελούνται από την ίδια ακολουθία στάσεων.

Transitions

Περιέχει όλα τα ζεύγη των στάσεων για τα οποία μπορεί να γίνει μετάβαση από μία διαδρομή σε μια άλλη και την απόσταση μεταξύ τους. Η υπό εξέταση γεωγραφική περιοχή χωρίζεται σε κελιά και για κάθε κελί καθώς και για τα γειτονικά του, βρίσκονται όλα τα ζευγάρια στάσεων που απέχουν λιγότερο από μια δεδομένη απόσταση. Στη συνέχεια τα δεδομένα φιλτράρονται και εμπλουτίζονται κατάλληλα ώστε να είναι διαθέσιμα προς χρήση από τον αλγόριθμο.

2.2 Απόσταση μεταξύ δύο σημείων

Για την εύρεση της απόστασης μεταξύ δύο σημείων με γνωστές γεωγραφικές συντεταγμένες χρησιμοποιήθηκε η εξίσωση haversine [11]. Η εξίσωση είναι η εξής:

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat_1) * \cos(lat_2) * \sin^2\left(\frac{\Delta long}{2}\right)$$

$$c = 2 * \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R * c$$

Όπου R η ακτίνα της Γης (6.371 km). Οι γωνίες είναι σε ακτίνια.

3. ΑΛΓΟΡΙΘΜΟΣ

3.1 Δεδομένα εισόδου

Ως δεδομένα του προβλήματος είναι οι γεωγραφικές συντεταγμένες των σημείων αναχώρησης και άφιξης καθώς και η ημέρα και ώρα αναχώρησης.

3.2 Διαδρομή

Το αποτέλεσμα της εκτέλεσης του αλγορίθμου είναι ένα ταξινομημένο σύνολο διαδρομών. Η ταξινόμηση γίνεται με βάση τη βαθμολογία που λαμβάνει η διαδρομή. Ο τρόπος βαθμολόγησης μιας διαδρομής περιγράφεται αναλυτικά σε επόμενη παράγραφο. Κάθε διαδρομή αποτελείται από ένα ή περισσότερα (το πολύ n) σκέλη. Κάθε σκέλος είναι υποσύνολο ενός δρομολογίου (trip) όπως αυτό ορίζεται παραπάνω. Ο αλγόριθμος δημιουργεί και επεξεργάζεται κατά την εκτέλεσή του σύνολα διαδρομών. Σε κάθε βήμα μία διαδρομή λαμβάνει μία από τις παρακάτω καταστάσεις:

- Εύρεση επόμενου σκέλους
- Ολοκληρωμένη
- Απορριφθείσα

Όσο μια διαδρομή έχει δεν έχει χαρακτηριστεί ως «ολοκληρωμένη» ή «απορριφθείσα» εξετάζεται σε κάθε βήμα του αλγορίθμου.

3.3 Ευρετικές Μέθοδοι (Heuristics)

Προκειμένου να αποτραπεί η εξαντλητική επεξεργασία όλων των πιθανών διαδρομών, να αποκλειστεί η εξέταση διαδρομών που δεν πληρούν ορισμένες βασικές περιοριστικές συνθήκες και να διασφαλιστεί ότι ο αλγόριθμος θα τερματιστεί μετά από μικρό αριθμό βημάτων, χρησιμοποιούνται δύο ευρετικές μέθοδοι. Αυτές οι μέθοδοι θέτουν ορισμένους περιορισμούς στα χαρακτηριστικά μιας διαδρομής προκειμένου αυτή να μπορεί να ανήκει στο σύνολο των πιθανών λύσεων του προβλήματος.

3.3.1 Κανόνας απόστασης

Έστω d η Ευκλείδεια απόσταση μεταξύ των σημείων αναχώρησης και προορισμού και σ το πλήθος των σκελών της υπό εξέταση διαδρομής. Για να πληροί η διαδρομή τον κανόνα απόστασης θα πρέπει για κάθε σκέλος να υπάρχει τουλάχιστο μία στάση η οποία να απέχει από το σημείο προορισμού απόσταση μικρότερη ή ίση από τις τιμές του παρακάτω πίνακα:

Σκέλος	Ελάχιστη απόσταση
1	-
2	$d / 2$
3	$d / 10$
4	$\min(d / 25, 300m)$

Οι παραπάνω τιμές προέκυψαν με εμπειρικό τρόπο. Μπορούν όμως να καθορίζονται δυναμικά, πχ ανάλογα με την έκταση της περιοχής που εξυπηρετεί το σύστημα συγκοινωνιών και την πυκνότητα των στάσεων σε αυτό.

3.3.2 Λοιποί περιορισμοί

Εφαρμόζονται οι εξής περιορισμοί:

1. Μια διαδρομή δεν μπορεί να έχει παραπάνω από n σκέλη.
2. Η απόσταση μεταξύ δύο στάσεων μετεπιβίβασης δεν πρέπει να υπερβαίνει ένα συγκεκριμένο όριο.
3. Ο χρόνος μετεπιβίβασης πρέπει να μην υπερβαίνει ένα συγκεκριμένο όριο.
4. Η απόσταση από το σημείο αναχώρησης στην κοντινότερη στάση δεν πρέπει να υπερβαίνει ένα συγκεκριμένο όριο.

3.4 Υπολογισμός χρόνου βαδίσματος

Χρησιμοποιείται η Ευκλείδεια απόσταση μεταξύ δύο σημείων πολλαπλασιασμένη με το συντελεστή 1.3 (ως εκτίμηση για τυχόν μη ευθεία πορεία). Επίσης γίνεται η παραδοχή ότι ο επιβάτης βαδίζει με μέση ταχύτητα 5 km/h.

3.5 Βαθμολόγηση διαδρομής

Χρησιμοποιείται στην εξάλειψη ενδιάμεσων αποτελεσμάτων τα οποία κρίνονται ως ασύμφορα για περαιτέρω αναζήτηση. Αποφεύγεται έτσι η εξαντλητική αναζήτηση όλων των ενδεχομένων. Η βαθμολόγηση γίνεται με βάση τον τύπο:

$$r = 100 * \frac{\frac{d - \delta}{d}}{t}$$

Όπου:

d : Η Ευκλείδεια απόσταση των σημείων αναχώρησης και προορισμού

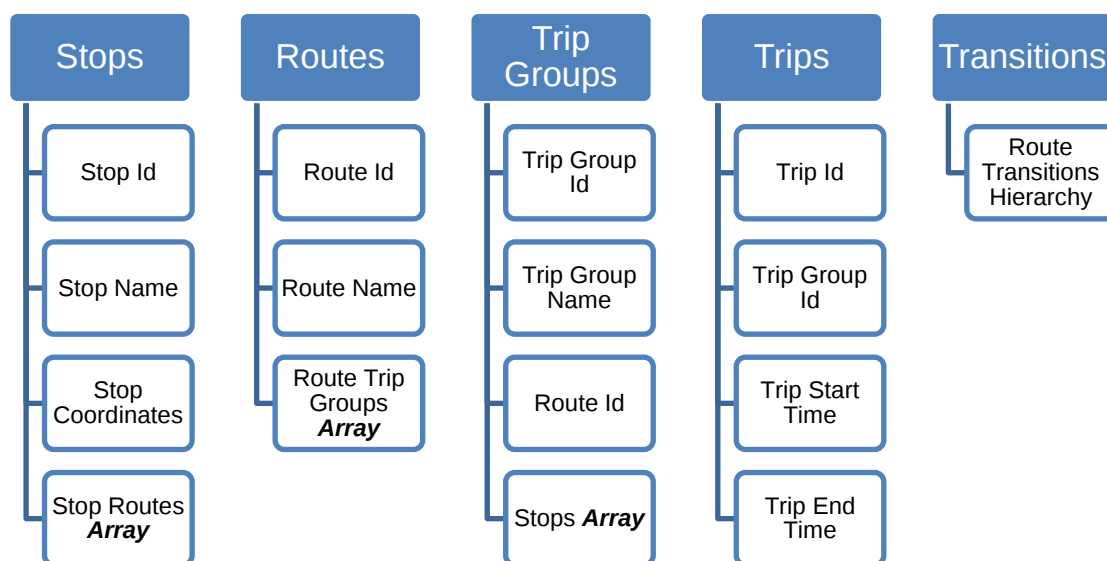
δ : Η Ευκλείδεια απόσταση της κοντινότερης στάσης (έστω σ_δ) του τελευταίου σκέλους της διαδρομής από το σημείο προορισμού.

t : Ο χρόνος που απαιτήθηκε για να φτάσει ο επιβάτης στη στάση σ_δ .

3.6 Περιγραφή εκτέλεσης του αλγορίθμου

3.6.1 Αρχικοποίηση της μνήμης

Κατά την εκκίνηση της εφαρμογής ανακτώνται από τη βάση και φορτώνονται σε κατάλληλες δομές στη μνήμη οι εξής πληροφορίες:



3.6.2 Εκτέλεση του αλγόριθμου

1^ο βήμα

Με δεδομένο τις γεωγραφικές συντεταγμένες του σημείου αναχώρησης βρίσκουμε τις κοντινότερες στάσεις σε αυτό. Για κάθε μία από τις στάσεις αυτές με βάση την ημέρα και ώρα αναχώρησης και την απόσταση από το αρχικό σημείο βρίσκουμε την ώρα άφιξης του επιβάτη στην 1^η στάση της διαδρομής.

2^ο βήμα

Για κάθε στάση και την ώρα άφιξης σε αυτή, ανακτούμε από τη βάση δεδομένων το πρώτο δρομολόγιο μεταξύ της ώρας άφιξης στη στάση και μιας ώρας αργότερα αν αυτό υπάρχει.

Έστω ότι ξεκινάμε από το σημείο x την ώρα t_0 . Στο 1^ο βήμα βρίσκουμε το σύνολο των κοντινών στάσεων $S = \{s_1, s_2, \dots, s_k\}$ με χρόνους άφιξης με βάση την Ευκλείδεια απόστασή τους από το σημείο x $\{t_1, t_2, \dots, t_k\}$. Στο 2^ο βήμα βρίσκουμε το 1^ο δρομολόγιο που διέρχεται από κάθε στάση s_n μεταξύ t_n και t_n+1h για κάθε ομάδα δρομολογίων τα οποία διέρχονται από τη στάση αυτή (αν υπάρχει τέτοιο δρομολόγιο).

Στο τέλος του 2^{ου} βήματος έχουμε ένα σύνολο διαδρομών που αποτελούνται από ένα μόνο δρομολόγιο. Κάθε δρομολόγιο διέρχεται από μια στάση που βρίσκεται κοντά στο σημείο x και σε κατάλληλη ώρα ώστε ο επιβάτης να έχει τον απαιτούμενο χρόνο να φτάσει από το σημείο αναχώρησης στη στάση.

3^ο βήμα

Για κάθε διαδρομή (η οποία ως τώρα αποτελείται από ένα και μόνο δρομολόγιο) που έχει προκύψει από την εκτέλεση του 2^{ου} βήματος βρίσκουμε την κοντινότερη (με βάση πάντα την Ευκλείδεια απόσταση) στάση στο σημείο προορισμού. Η διαδρομή βαθμολογείται με βάση το χρόνο και το ποσοστό της Ευκλείδειας απόστασης που έχει καλυφθεί μεταξύ του σημείου αναχώρησης και προορισμού.

Αν η στάση αυτή είναι αρκετά κοντά στο σημείο προορισμού (με βάση τη στατική παράμετρο που έχει οριστεί) η διαδρομή σημειώνεται ως ολοκληρωμένη. Σε διαφορετική περίπτωση σημειώνεται προς περαιτέρω επεξεργασία.

Οι διαδρομές που προκύπτουν από αυτό το βήμα αποθηκεύονται σε κατάλληλη δομή στη μνήμη προκειμένου να τύχουν επεξεργασίας από το επόμενο βήμα.

4^ο βήμα

Το βήμα αυτό εκτελείται έως ότου όλες οι διαδρομές έχουν σημειωθεί ως ολοκληρωμένες ή έχουν απορριφθεί.

Για κάθε διαδρομή που έχει προκύψει από το 3^ο βήμα ή για τις κ διαδρομές με την καλύτερη βαθμολογία από προηγούμενη εκτέλεση του 4^{ου} βήματος γίνονται τα εξής:

- Αναζητούνται όλες οι ομάδες δρομολογίων που διασταυρώνονται με το τελευταίο δρομολόγιο της διαδρομής, δηλαδή είτε χρησιμοποιούν κάποια κοινή στάση είτε διέρχονται από κάποια κοντινή στάση. Για το σκοπό αυτό χρησιμοποιείται ο πίνακας transitions.
- Για κάθε τέτοια ομάδα δρομολογίων δημιουργείται μια νέα διαδρομή αντίγραφο αυτής την οποία εξετάζουμε στην οποία προστίθεται ως νέο σκέλος ένας αντιπρόσωπος της ομάδας αυτής.
- Για το νέο σκέλος βρίσκουμε την κοντινότερη στάση στο σημείο προορισμού **καθώς και όλες τις στάσεις που ανήκουν σε αυτό και ικανοποιούν τον κανόνα απόστασης**. Εάν το νέο σκέλος της διαδρομής δεν έχει καμία στάση που να ικανοποιεί τον κανόνα απόστασης τότε η διαδρομή απορρίπτεται, διαφορετικά προχωρούμε στα επόμενα βήματα.
- Εξετάζουμε αν υπάρχει χρονική συνάφεια μεταξύ του τελευταίου δρομολογίου της διαδρομής και έστω ενός από αυτά που ανήκουν στην ομάδα του σκέλους που προστέθηκε. Αν δεν υπάρχει χρονική συνάφεια τότε η διαδρομή απορρίπτεται, διαφορετικά προχωρούμε στα επόμενα βήματα. *Πχ αν το τελευταίο σκέλος έχει ώρα εκκίνησης 13:40 και ώρα τερματισμού 15:10 ενώ κανένα από τα δρομολόγια της ομάδας στην οποία ανήκει το νέο σκέλος της διαδρομής δεν εκτελούνται μέσα ή κοντά στα άκρα του παραπάνω χρονικού διαστήματος θεωρούμε ότι δεν υπάρχει η επιθυμητή χρονική συνάφεια.*
- Ανακτούμε από τη βάση όλα τα δρομολόγια που ανήκουν στην ομάδα του νέου σκέλους και βρίσκουμε το κατάλληλο δρομολόγιο καθώς και τις στάσεις μεταξύ των οποίων θα πραγματοποιηθεί η μετεπιβίβαση.
- Βαθμολογούμε τη διαδρομή που έχει προκύψει από την εκτέλεση των προηγούμενων βημάτων και εξετάζουμε:
 - Αν το τελευταίο σκέλος της διαδρομής φτάνει αρκετά κοντά στο σημείο προορισμού, οπότε και σημειώνουμε τη διαδρομή ως ολοκληρωμένη.

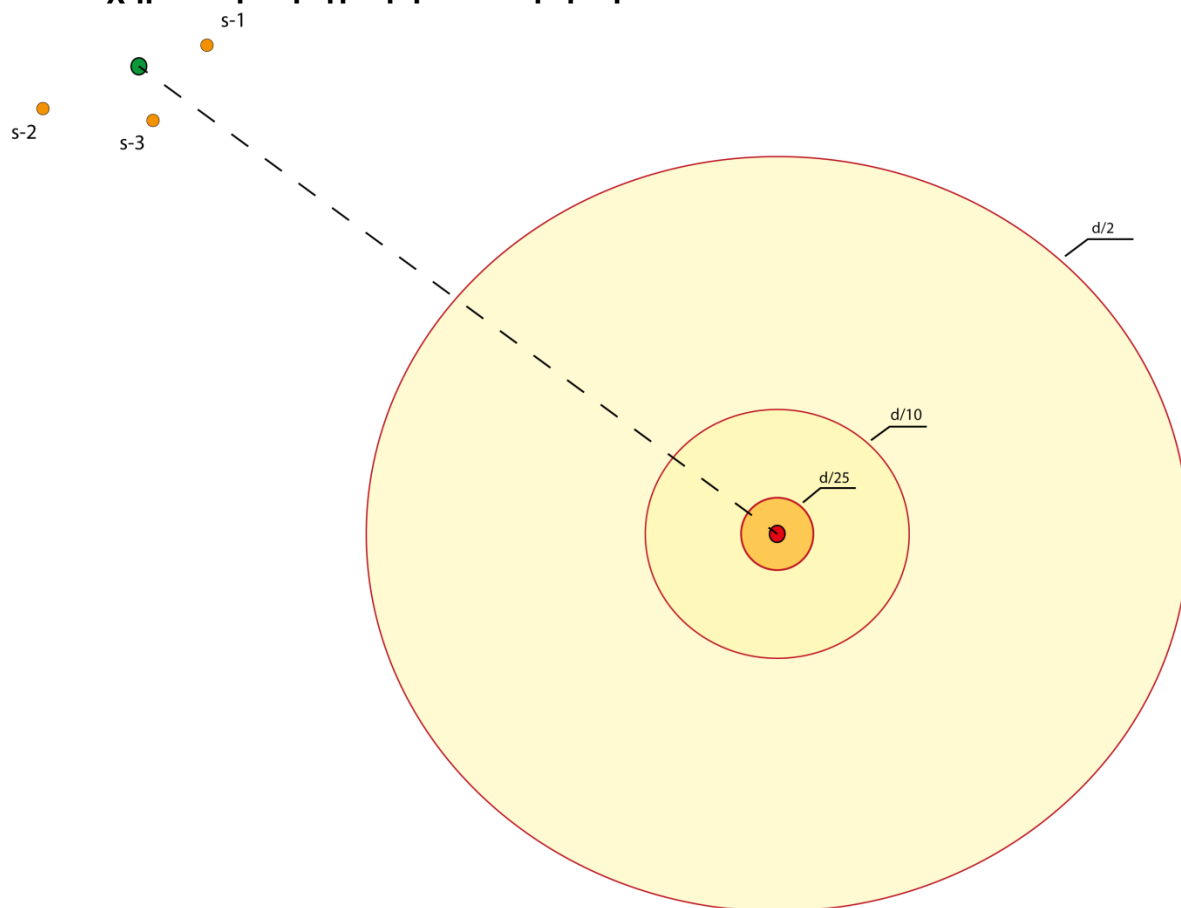
- Αν δεν ισχύει η παραπάνω συνθήκη και οι μετεπιβιβάσεις της διαδρομής είναι λιγότερες από το όριο που έχει τεθεί, η διαδρομή σημειώνεται για την εύρεση επόμενου σκέλους.
- Αν δεν ισχύει καμία από τις παραπάνω δύο συνθήκες η διαδρομή σημειώνεται ως απορριφθείσα.

Στο τέλος του 4^{ου} βήματος έχουμε ένα σύνολο βαθμολογημένων διαδρομών, κάποιες από τις οποίες πιθανόν να έχουν ολοκληρωθεί. Για τις n -βέλτιστες διαδρομές οι οποίες έχουν σημειωθεί για περαιτέρω επεξεργασία επαναλαμβάνουμε το βήμα αυτό.

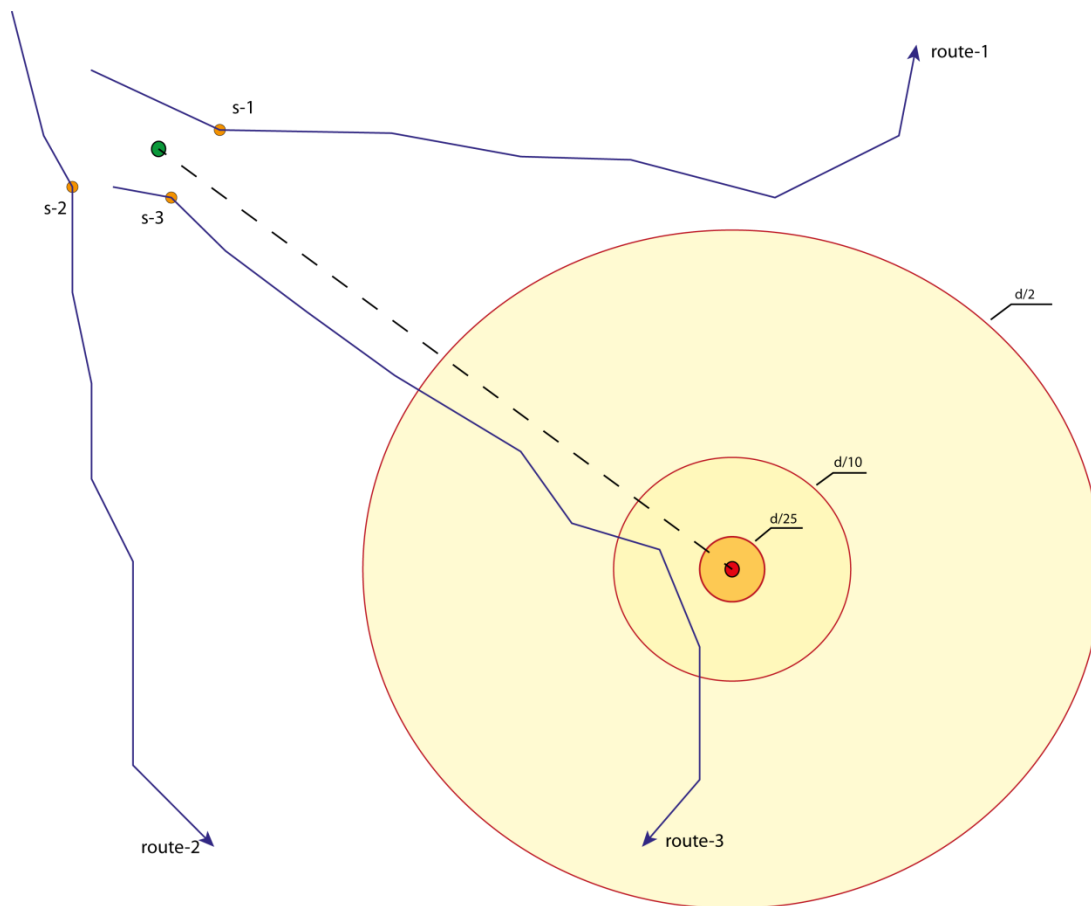
Ο αλγόριθμος θα ολοκληρωθεί μόλις δεν υπάρχουν πλέον διαδρομές προς επεξεργασία. Αυτό θα συμβεί μετά από το πολύ n επαναλήψεις του 4^{ου} βήματος όπου n το μέγιστο επιτρεπόμενο πλήθος μετεπιβιβάσεων.

Εφόσον πληρούται η συνθήκη εξόδου από το 4^ο βήμα, οι διαδρομές που έχουν σημειωθεί ως ολοκληρωμένες ταξινομούνται ως προς τη βαθμολογία και απεικονίζεται η βέλτιστη.

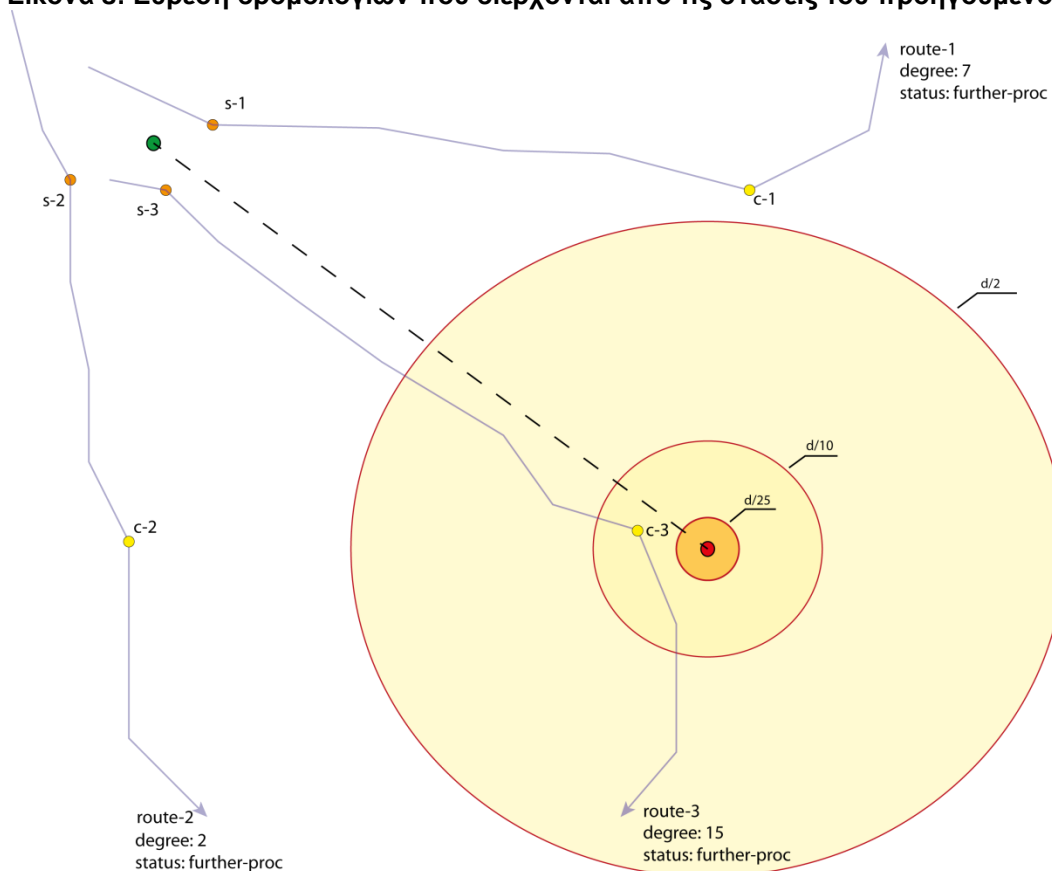
3.6.3 Σχηματική περιγραφή του αλγόριθμου



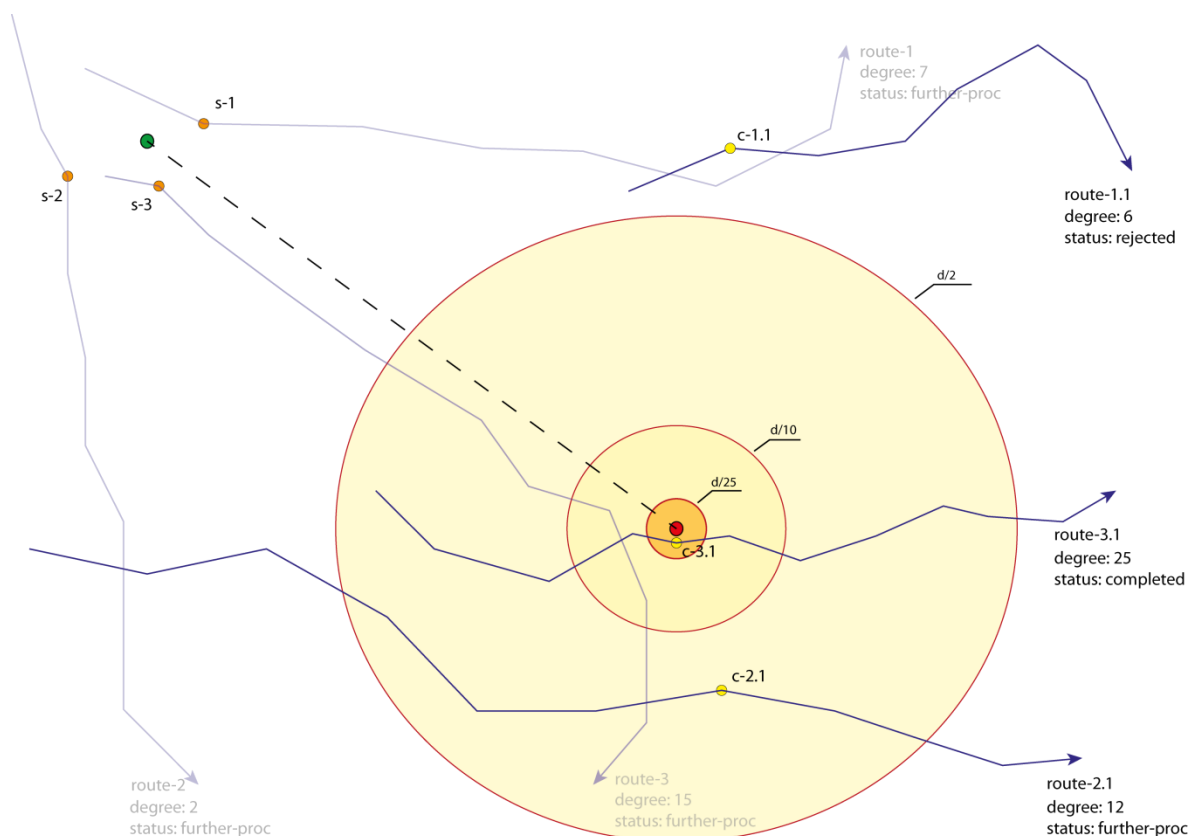
Εικόνα 2: Εύρεση κοντινών στάσεων στο σημείο αναχώρησης. (στάσεις s_1 , s_2 και s_3)



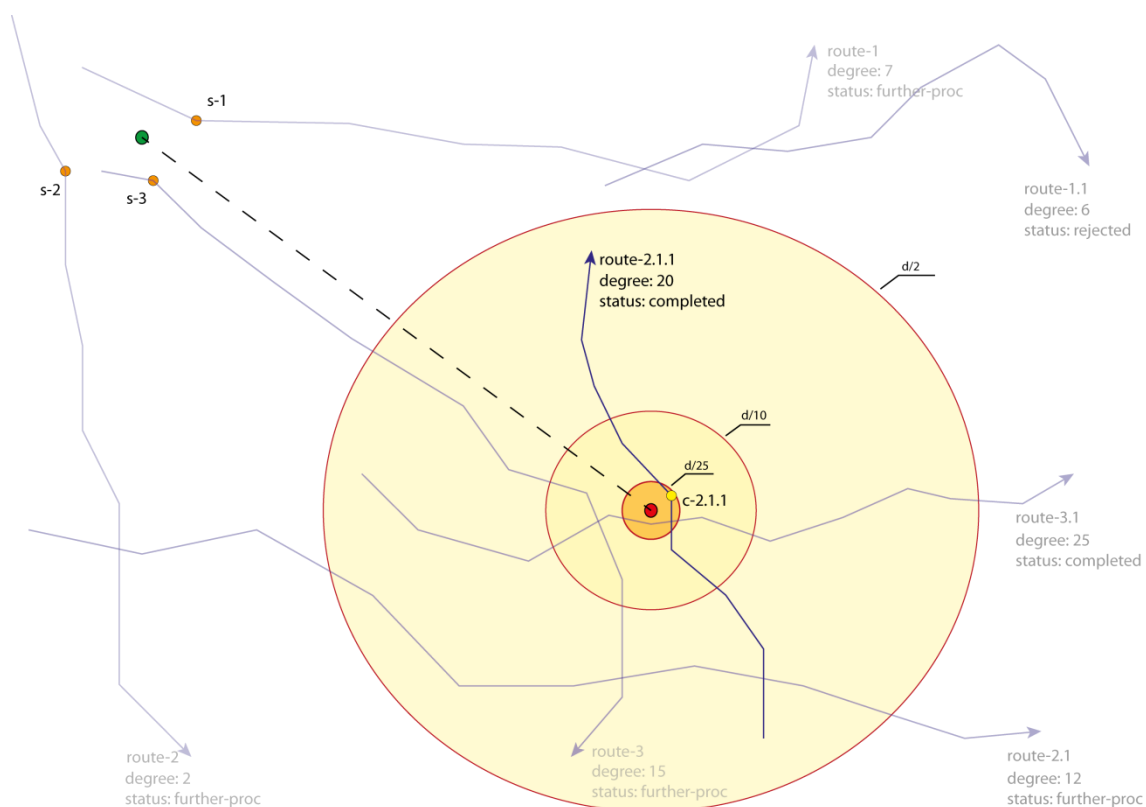
Εικόνα 3: Εύρεση δρομολογίων που διέρχονται από τις στάσεις του προηγούμενου βήματος.



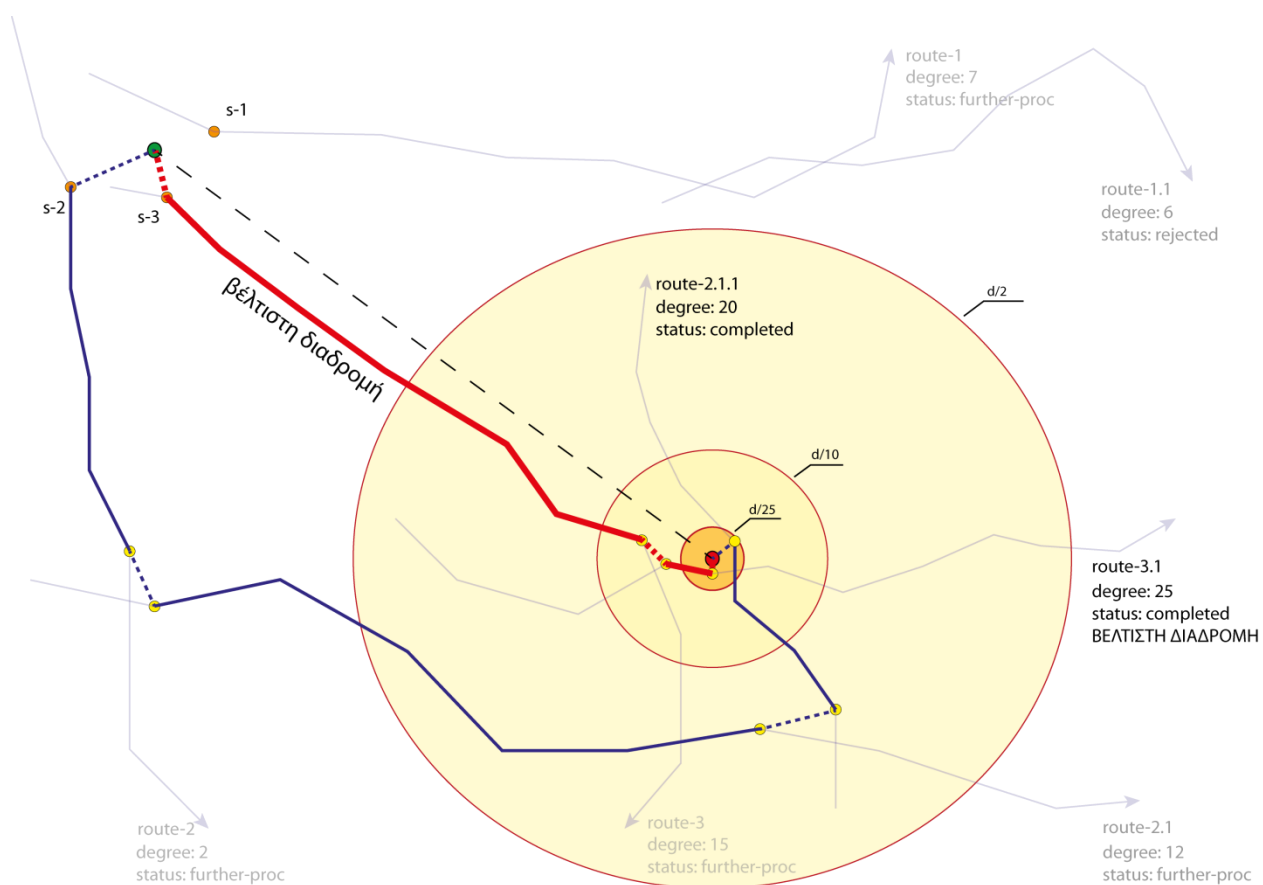
Εικόνα 4: Εύρεση της κοντινότερης στάσης στο σημείο προορισμού για κάθε διαδρομή. Βαθμολόγηση των διαδρομών με βάση το ποσοστό της απόστασης και το χρόνο στον οποίο αυτό έχει καλυφθεί.



Εικόνα 5: Προσθήκη νέου σκέλους σε κάθε διαδρομή και εξέταση / βαθμολόγηση των δρομολογίων που προκύπτουν.



Εικόνα 6: Επανάληψη του προηγούμενου βήματος για όσα δρομολόγια είναι εφικτό.



Εικόνα 7: Η επεξεργασία των διαδρομών έχει ολοκληρωθεί. Κατάταξη και απεικόνιση της βέλτιστης διαδρομής.

4. ΥΛΟΠΟΙΗΣΗ

Για την επίδειξη του αλγόριθμου υλοποιήθηκε μια εφαρμογή η οποία μπορεί να δέχεται στοιχεία από το χρήστη και να παρουσιάζει το αποτέλεσμα της αναζήτησης. Μπορεί να γίνει αναζήτηση δρομολογίων για τις πόλεις της Αθήνας και του San Francisco.

4.1 Αρχιτεκτονική

Το interface της εφαρμογής έχει υλοποιηθεί σε HTML / CSS (έχουν χρησιμοποιηθεί με κατάλληλη προσαρμογή ορισμένα έτοιμα στοιχεία από το internet όπως το ημερολόγιο και μέρος του CSS αρχείου).

Ο αλγόριθμος δρομολόγησης υλοποιήθηκε σε JavaScript. Για τις ανάγκες τις αποθήκευσης, επεξεργασίας και ανάκτησης των δεδομένων των δρομολογίων χρησιμοποιήθηκε μια βάση SQLite, η αλληλεπίδραση με την οποία γίνεται μέσω JavaScript APIs [12]. Τα εν λόγω APIs αποτελούν επέκταση του JavaScript engine και υποστηρίζονται μόνο από browsers συμβατούς με το WebKit framework [13]. Η υλοποίηση είναι προς το παρόν συμβατή μόνο με τον Safari 4+. [14]

Τα δεδομένα των δρομολογίων των συστημάτων συγκοινωνιών είναι διαθέσιμα σε comma delimited ASCII αρχεία, συμβατά με το πρότυπο GTFS.

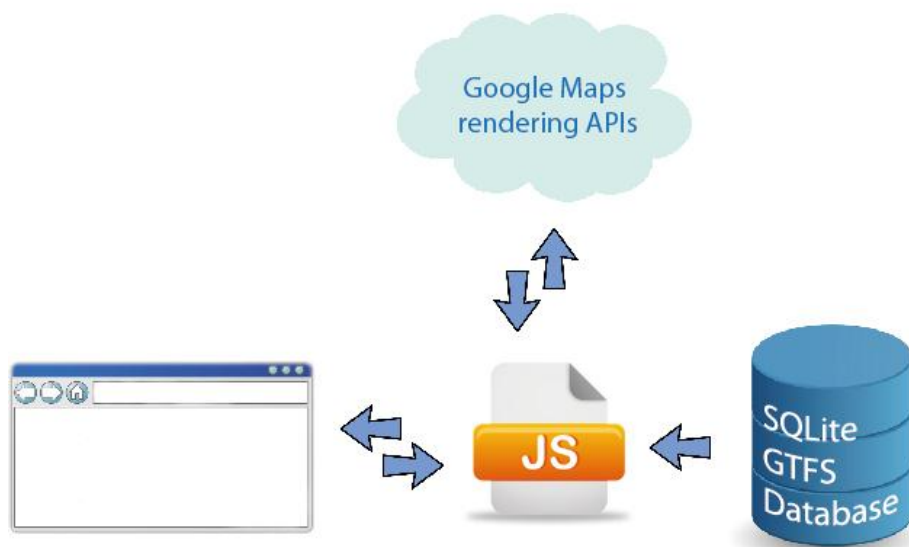
Η βάση δεδομένων της εφαρμογής τροφοδοτείται με τα στοιχεία του συστήματος συγκοινωνιών με τη χρήση JavaScript functions που έχουν γραφτεί για το σκοπό αυτό. Οι διαδικασίες αυτές αφενός εισάγουν τα δεδομένα των ASCII αρχείων, αφετέρου εκτελούν την προ-επεξεργασία που απαιτείται για τη δημιουργία ορισμένων πρόσθετων πινάκων που είναι απαραίτητοι στον αλγόριθμο.



Εικόνα 8:
Επεξεργασία
δεδομένων
GTFS.

Για την εισαγωγή του σημείου αναχώρησης και προορισμού αλλά και για την απεικόνιση των αποτελεσμάτων χρησιμοποιείται το Google Maps [15].

Μέσω του web interface ο χρήστης επιλέγει στο χάρτη τα σημεία αναχώρησης και προορισμού. Επιλέγει επίσης την επιθυμητή ημέρα και ώρα αναχώρησης. Μέσω των Google Maps APIs ανακτώνται οι γεωγραφικές συντεταγμένες των σημείων αναχώρησης και προορισμού και μαζί με την ώρα αναχώρησης τροφοδοτούν τον αλγόριθμο δρομολόγησης. Ο αλγόριθμος βάσει των ορισμάτων και των δεδομένων δρομολογίων επιστρέφει ένα σύνολο από προτεινόμενες διαδρομές ταξινομημένες ως προς τον απαιτούμενο χρόνο. Η βέλτιστη διαδρομή απεικονίζεται στο χάρτη με τη χρήση κατάλληλων Google Maps APIs.



Εικόνα 9: Αρχιτεκτονική της εφαρμογής.

4.2 Παραδείγματα

Ο αλγόριθμος μπορεί να χρησιμοποιηθεί για κάθε πόλη για την οποία είναι διαθέσιμα δεδομένα του συστήματος συγκοινωνιών της στο πρότυπο GTFS. Το μόνο που χρειάζεται (με σκοπό να απαλειφθεί στο μέλλον) είναι ρύθμιση ορισμένων ευρετικών κανόνων όπως του κανόνα απόστασης ο οποίος καθορίζει τον τρόπο σύγκλισης των δρομολογίων στο σημείο προορισμού καθώς και του μέγιστου επιτρεπόμενου αριθμού σκελών μιας διαδρομής. Οι πόλεις για τις οποίες εφαρμόστηκε ο αλγόριθμος είναι της Αθήνας και του San Francisco. Στον παρακάτω πίνακα παρουσιάζονται ορισμένα συγκριτικά στοιχεία για τα συστήματα συγκοινωνιών των δύο πόλεων.

San Francisco

Γραμμές	82
Στάσεις	4.411
Δρομολόγια	27.509
Αφίξεις / Αναχωρήσεις	1.122.271
Μέγεθος βάσης	95 Mbytes

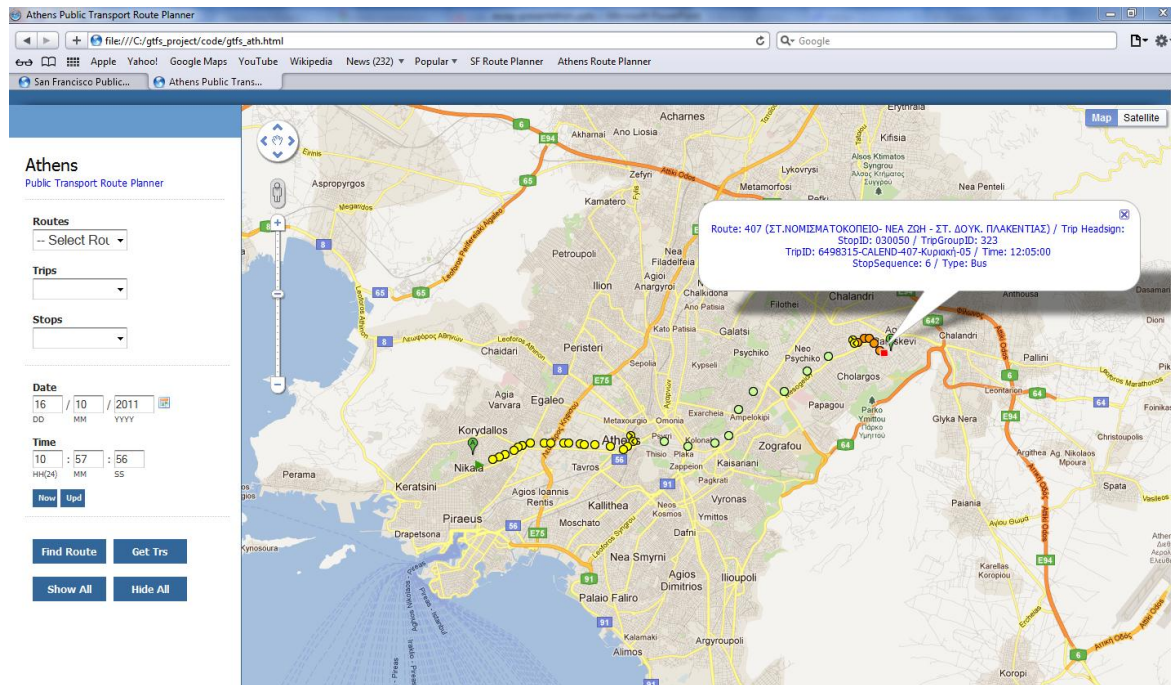
Αθήνα

Γραμμές	295
Στάσεις	7.778
Δρομολόγια	59.027
Αφίξεις / Αναχωρήσεις	2.248.284
Μέγεθος βάσης	546 Mbytes

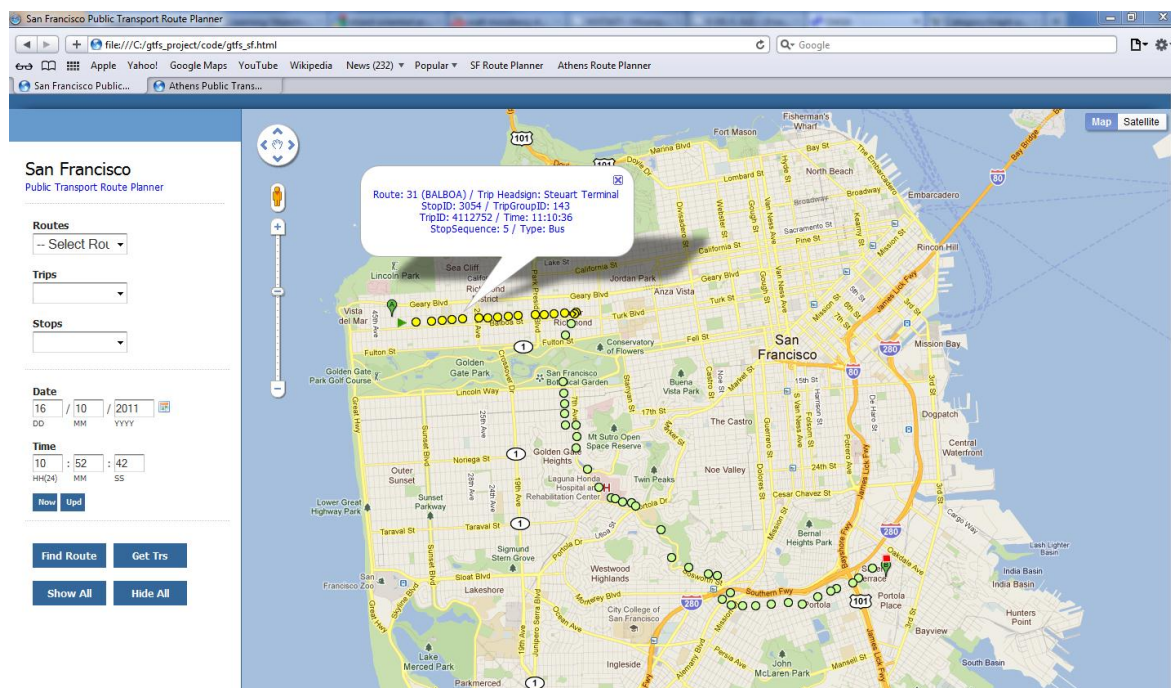
Εύρεση βέλτιστων δρομολογίων σε σύστημα αστικών συγκοινωνιών με βάση το πρότυπο GTFS

Η ταχύτητα εκτέλεσης του αλγόριθμου εξαρτάται από το μέγεθος της βάσης δεδομένων γιατί απαιτείται τουλάχιστο μια αναζήτηση στη βάση για κάθε ομάδα δρομολογίων. Για το λόγο αυτό η απόδοση του αλγόριθμου για την Αθήνα είναι ελαφρώς χειρότερη από αυτή για το San Francisco.

Στις παρακάτω εικόνες φαίνονται από μία αναζήτηση στην Αθήνα και το San Francisco.



Εικόνα 10: Αναζήτηση δρομολογίου στην Αθήνα.



Εικόνα 11: Αναζήτηση δρομολογίου στο San Francisco.

4.3 Εργαλεία υλοποίησης

Για την υλοποίηση του αλγορίθμου και του γραφικού περιβάλλοντος χρησιμοποιήθηκαν τα εξής εργαλεία:

- PSpad Editor (συγγραφή JavaScript κώδικα)
- Safari 5.1 JavaScript Console (εκτέλεση / debugging JavaScript - HTML)
- SQLite Manager Mozilla Plug-in (ανάλυση / επεξεργασία δεδομένων GTFS)
- Adobe Dreamweaver CS5 (συγγραφή / επεξεργασία HTML – CSS)

5. ΜΕΛΛΟΝΤΙΚΕΣ ΒΕΛΤΙΩΣΕΙΣ

Προκειμένου η υλοποίηση να είναι πιο αποτελεσματική απαιτούνται ορισμένες βελτιώσεις που αφορούν στην απόδοση, αξιοπιστία και φορητότητα του αλγόριθμου και της εφαρμογής που τον χρησιμοποιεί.

5.1 Σμίκρυνση βάσης δεδομένων GTFS

Η πρώτη βελτίωση αφορά στο μέγεθος της βάσης δεδομένων που χρησιμοποιείται για τα στοιχεία των συστημάτων συγκοινωνιών και σχετίζεται κυρίως με την απόδοση του αλγόριθμου. Τα δεδομένα των συστημάτων συγκοινωνιών σε ASCII μορφή είναι 50 και 150 Mbytes για το San Francisco και την Αθήνα αντίστοιχα ενώ το μέγεθος αυτό εκτινάσσεται στα 95 και 545 Mbytes για τις βάσεις SQLite οι οποίες χρησιμοποιούνται για τον αλγόριθμο επεξεργασίας τους. Τα μεγέθη αυτά είναι απαγορευτικά (ή μη πρακτικά) για τη χρήση του αλγόριθμου σε φορητές συσκευές (κυρίως smartphones). Επίσης το μέγεθος της βάσης είναι επιβαρυντικός παράγοντας για την απόδοση του αλγόριθμου λόγω της αυξανόμενης χρονικής υστέρησης που σχετίζεται με την εκτέλεση ερωτημάτων στη βάση.

Η συντριπτική πλειοψηφία των δεδομένων των συστημάτων συγκοινωνιών αφορούν στις αναχωρήσεις / αφίξεις των συγκοινωνιακών μέσων στις στάσεις του συστήματος. Αντί να χρειάζεται να καταγράφεται κάθε αναχώρηση άφιξη για κάθε δρομολόγιο, θα μπορούσε με κατάλληλη επεξεργασία (και ενδεχόμενη κβαντοποίηση) των δεδομένων αναχώρησης / άφιξης να προκύψει ένα ή περισσότερα χρονικά πρότυπα για κάθε ομάδα δρομολογίων, βάσει των οποίων θα μπορούσε να παραχθεί ολόκληρο το πρόγραμμα.

5.2 Προτίμηση μέσων σταθερής τροχιάς

Ο αλγόριθμος χειρίζεται ισότιμα όλα τα συγκοινωνιακά μέσα, είτε πρόκειται για μέσα σταθερής τροχιάς είτε όχι. Αυτό μπορεί να οδηγήσει σε αποτελέσματα τα οποία στην πράξη δεν θα είναι βέλιστα για το λόγο ότι το πρόγραμμα δρομολογίων ενός μέσου σταθερής τροχιάς είναι πολύ πιθανότερο να τηρηθεί απ' ό,τι το πρόγραμμα ενός συμβατικού μέσου. Για το λόγο αυτό θα μπορούσε ο αλγόριθμος να προτιμά διαδρομές που πραγματοποιούνται εξ ολοκλήρου ή εν μέρει με μέσα σταθερής τροχιάς αν η αξιολόγησή τους είναι πολύ κοντά σε αυτή των διαδρομών που πραγματοποιούνται με συμβατικά μέσα, παρέχοντας έτσι αποτελέσματα τα οποία θα είναι πιθανόν πιο έγκυρα από αυτά που δίνει στην τρέχουσα μορφή του.

5.3 Δυναμικές ευρετικές μέθοδοι

Ο αλγόριθμος χρησιμοποιεί ορισμένες στατικές ευρετικές μεθόδους προκειμένου να αποκλείει ενδιάμεσα αποτελέσματα τα οποία δεν θα οδηγήσουν (με μεγάλη πιθανότητα) σε λύση, προκειμένου να ελαφρύνει το φόρτο των απαιτούμενων υπολογισμών.

Ορισμένες από αυτές τις μεθόδους είναι συνάρτηση σύγκλισης της διαδρομής στο σημείο προορισμού, ο μέγιστος χρόνος που πρέπει να μεσολαβεί κατά τη μετάβαση από ένα μέσο σε ένα άλλο και ο καθορισμός του μέγιστου πλήθους σκελών (δρομολογίων) από τα οποία μπορεί να αποτελείται μία διαδρομή. Προς το παρόν οι παράμετροι των δύο αυτών κανόνων καθορίστηκαν με εμπειρικό τρόπο για κάθε μία από τις δύο πόλεις για τις οποίες δοκιμάστηκε ο αλγόριθμος. Σημαντικό σημείο βελτίωσης είναι η δημιουργία μεθόδων οι οποίες βάσει των χαρακτηριστικών του συστήματος συγκοινωνιών (πχ γεωγραφική κατανομή στάσεων, χρονική και γεωγραφική κατανομή των δρομολογίων) να καθορίζουν τις παραμέτρους για κάθε μία από τις προηγούμενες ευρετικές μεθόδους που επηρεάζουν σημαντικά τα αποτελέσματα και την ταχύτητα εκτέλεσης του αλγόριθμου.

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

GTFS	General Transit Feed Specification
HTML	HyperText Markup Language
CSS	Cascading Style Sheets

ΑΝΑΦΟΡΕΣ

- [1] R. Bellman, «On a routing problem,» *Quarterly of Applied Mathematics*, τόμ. 16, αρ. 1, pp. 87-90, 1958.
- [2] E. W. Dijkstra, «A Note on Two Problems in Connection with Graphs,» *Numerische Mathematik*, τόμ. 1, αρ. 1, pp. 269-271, 1959.
- [3] P. E. Hart, N. J. Nilsson και B. Raphael, «A Formal Basis for the Heuristic Determination of Minimum Cost Paths,» *Ieee Transactions On Systems Science And Cybernetics*, τόμ. 4, αρ. 2, pp. 100-107, 1968.
- [4] R. Huang, «A Schedule-based Pathfinding Algorithm for Transit Networks Using Pattern First Search,» *Geoinformatica*, τόμ. 11, αρ. 2, pp. 269-285, 2007.
- [5] M. Fragouli και A. Delis, «EasyTransport: An Effective Navigation and Transportation Guide for Wide Geographic Areas,» σε *14th IEEE International Conference on Tools with Artificial Intelligence*, 2002.
- [6] B. Ding, J. Yu και L. Qin, «Finding time-dependent shortest paths over large graphs,» New York, USA, 2008.
- [7] R. Huang και Z.-R. Peng, «A spatiotemporal data model for dynamic transit networks,» *International Journal of Geographical Information Science*, τόμ. 22, pp. 527-545, 2008.
- [8] O. Wolfson και B. Xu, «Spatio-temporal Databases in Urban Transportation,» *Data Engineering*, 2010.
- [9] Google, «Google Code,» [Ηλεκτρονικό]. Available: http://code.google.com/intl/el-GR/transit/spec/transit_feed_specification.html.
- [10] «GTFS Data Exchange,» [Ηλεκτρονικό]. Available: <http://www.gtfs-data-exchange.com/agency/san-francisco-municipal-transportation-agency/>.
- [11] Movable Type Ltd, «Movable Type - Information Design & Management,» [Ηλεκτρονικό]. Available: <http://www.movable-type.co.uk/scripts/latlong.html>.
- [12] W3C, «W3C,» [Ηλεκτρονικό]. Available: <http://www.w3.org/TR/webdatabase/>.
- [13] WebKit, «The WebKit Open Source Project,» [Ηλεκτρονικό]. Available: <http://www.webkit.org/>.
- [14] Apple Inc., «Safari Developer Library,» [Ηλεκτρονικό]. Available: http://developer.apple.com/library/safari/#documentation/iPhone/Conceptual/SafariJSDatabaseGuide/UsingtheJavascriptDatabase/UsingtheJavascriptDatabase.html#//apple_ref/doc/uid/TP40007256-CH3-SW1.
- [15] Google, «Google Code,» [Ηλεκτρονικό]. Available: <http://code.google.com/intl/el-GR/apis/maps/documentation/javascript/reference.html>.