



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ, ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ & ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

**Αλγοριθμικά Ζητήματα Κινητικότητας
σε Έξυπνες Πόλεις**

Καλλιόπη Γιαννακοπούλου

ΑΜ: 526

Πάτρα, Σεπτέμβριος 2017

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ, ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

© 2017 Καλλιόπη Γιαννακοπούλου — Με την επιφύλαξη παντός δικαιώματος

Στον Χρήστο, στην Μαριέττα και στον Πέτρο.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τους ανθρώπους που με στήριξαν και με βοήθησαν κατά τη διάρκεια της εκπόνησης αυτής της διδακτορικής διατριβής.

Καταρχήν, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Σωτήρη Νικολετσέα, για την εμπιστοσύνη, τις εποικοδομητικές του συμβουλές και την άψογη συνεργασία που είχαμε από τον Απρίλιο 2014 που ξεκίνησα την εκπόνηση της διδακτορικής μου διατριβής. Επίσης, ευχαριστώ τα μέλη της τριμελούς επιτροπής κ. Χρήστο Μακρή και κ. Σπύρο Σιούτα, για τις εποικοδομητικές τους παρατηρήσεις, καθώς και τα μέλη της 7-μελούς εξεταστικής επιτροπής για την τιμή που μου έκαναν να συμμετάσχουν στην εξέταση της διδακτορικής μου διατριβής.

Θα ήθελα επίσης να ευχαριστήσω όλα τα μέλη του εργαστηριακού χώρου B020 για την πολύτιμη βοήθειά τους σε ότι και αν χρειάστηκα, και ειδικά τον Ανδρέα Παρασκευόπουλο για την εποικοδομητική συνεργασία που είχαμε. Θέλω ακόμη να ευχαριστήσω τους Alessio Cionini, Gianlorenzo D' Angelo, Mattia D'Emidio, Daniele Frigioni από το Πανεπιστήμιο L'Aquila της Ιταλίας για την επιτυχημένη ερευνητική συνεργασία που είχαμε.

Ευχαριστώ το Πανεπιστήμιο Πατρών και συγκεκριμένα το Τμήμα Μηχανικών Η/Υ & Πληροφορικής για την δυνατότητα που μου έδωσαν να εκπονήσω την διδακτορική μου διατριβή. Ευχαριστώ το ΙΤΥΕ «Διόφαντος» για την δυνατότητα που μου έδωσε να συμμετέχω σε ένα πολύ ενδιαφέρον ερευνητικό πρόγραμμα (MOVESMART) καθώς και για την οικονομική υποστήριξη που μου παρείχε.

Ένα μεγάλο ευχαριστώ, τέλος, το οφείλω στο σύζυγό και στα παιδιά μου, για την συμπαράστασή τους, την αγάπη τους, αλλά και για το χρόνο που έλειψα από κοντά τους προκειμένου να ολοκληρώσω την διδακτορική μου διατριβή.

Περίληψη

Τα τελευταία χρόνια δίνεται συνεχώς όλο και μεγαλύτερη έμφαση στις προσωποποιημένες υπηρεσίες ταξιδιωτικής πλοήγησης και ανανεωνόμενης κινητικότητας μέσα σε έξυπνες πόλεις. Οι υπηρεσίες αυτές συνδυάζουν Μέσα Μαζικής Μεταφοράς (MMM) και μικρά ηλεκτρικά αυτοκίνητα ή ποδήλατα, χρησιμοποιώντας τη λεγόμενη τεχνική του πληθοπορισμού για συλλογή δεδομένων κίνησης πραγματικού χρόνου και αξιολόγησης προτεινόμενων διαδρομών. Στόχος είναι η δημιουργία ενός συστήματος που θα επιτρέπει τη γρήγορη εύρεση βέλτιστων διαδρομών.

Οι βασικότερες προκλήσεις στο σχεδιασμό ενός τέτοιου συστήματος είναι τεχνολογικές και αλγοριθμικές. Η τεχνολογική πρόκληση αφορά στη συλλογή, αποθήκευση, διαχείριση, και ενημέρωση ενός τεράστιου όγκου συγκοινωνιακών δεδομένων που είναι συνήθως χρονοεξαρτώμενα, και η παροχή (μέσω αυτών) προσωποποιημένων υπηρεσιών ανανεωνόμενης κινητικότητας σε έξυπνες φορητές συσκευές. Αυτή η πρόκληση αντιμετωπίζεται συνήθως με τη δημιουργία ενός περιβάλλοντος υπολογιστικού νέφους, το οποίο αφενός θα υποστηρίζει την αποθήκευση, διαχείριση, και ενημέρωση των δεδομένων και αφετέρου θα φροντίζει για την παροχή πληροφοριών στις εφαρμογές των έξυπνων φορητών συσκευών για εύρεση βέλτιστων διαδρομών.

Η αλγοριθμική πρόκληση αφορά στην ανάπτυξη καινοτόμων αλγορίθμων για την παροχή αποδοτικών υπηρεσιών ταξιδιωτικής πλοήγησης σε έξυπνες φορητές συσκευές, με δεδομένα που θα λαμβάνουν από το υπολογιστικό νέφος. Οι υπηρεσίες αυτές εξασφαλίζουν την εύρεση ρεαλιστικών και χρήσιμων διαδρομών, αλλά και την άμεση παροχή ενημερωμένων διαδρομών στους συνδεδεμένους χρήστες, σε περίπτωση καθυστερήσεων, για την επικαιροποίηση του αρχικού πλάνου τους. Σκοπός είναι η δημιουργία μιας αλγοριθμικής βάσης για την υποστήριξη υπηρεσιών κινητικότητας σε αστικά συγκοινωνιακά περιβάλλοντα με χρήση MMM. Τα συστήματα MMM, που λειτουργούν βάσει συγκεκριμένου πίνακα δρομολογίων, πρέπει όχι μόνο να ανταποκρίνονται σε πραγματικό χρόνο σε ερωτήματα χρηστών για εύρεση βέλτιστων διαδρομών,

αλλά και να προβαίνουν σε ενημερώσεις των αποθηκευμένων συγκοινωνιακών πληροφοριών προκειμένου να ανταποκρίνονται (επίσης σε πραγματικό χρόνο) στις συχνές αλλαγές των προκαθορισμένων δρομολογίων λόγω καθυστερήσεων.

Τα βασικότερα αλγοριθμικά ζητήματα κινητικότητας και ταξιδιωτικής πλοήγησης (εύρεσης βέλτιστων διαδρομών ως προς ορισμένα κριτήρια) σε συστήματα MMM αφορούν στα θεμελιώδη προβλήματα της νωρίτερης άφιξης (NA - μετακίνηση ενός επιβάτη από ένα σταθμό S σε ένα σταθμό T , στον ελάχιστο δυνατό χρόνο), του ελάχιστου αριθμού μετεπιβιβάσεων (EAM - μετακίνηση ενός επιβάτη από ένα σταθμό S σε ένα σταθμό T , με τον ελάχιστο αριθμό μετεπιβιβάσεων), και της δυναμικής ενημέρωσης της πληροφορίας των πινάκων δρομολογίων σε περίπτωση καθυστερήσεων. Τα προβλήματα NA και EAM έχουν εκτενώς μελετηθεί στη βιβλιογραφία κάτω από δύο προσεγγίσεις: τη *διανυσματοκεντρική μοντελοποίηση* (όπου οι πίνακες δρομολογίων αναπαριστώνται ως διανύσματα) και τη *γραφοκεντρική μοντελοποίηση* (όπου οι πίνακες δρομολογίων αναπαριστώνται ως γραφήματα). Προηγούμενες πειραματικές μελέτες είχαν δείξει ότι οι διανυσματοκεντρικές προσεγγίσεις είναι γρηγορότερες όσον αφορά το χρόνο απάντησης των ερωτημάτων από τις γραφοκεντρικές. Από την άλλη πλευρά, οι διανυσματοκεντρικές προσεγγίσεις δεν έχουν μελετηθεί ως προς την αποδοτική ενημέρωση του πίνακα δρομολογίων σε περίπτωση καθυστερήσεων.

Στα πλαίσια της διδακτορικής διατριβής αναπτύχθηκαν νέες γραφοκεντρικές μέθοδοι που επιλύουν αποδοτικά τα προαναφερθέντα θεμελιώδη αλγοριθμικά προβλήματα κινητικότητας σε αστικά περιβάλλοντα με χρήση MMM, καθώς και μια εφαρμογή ταξιδιωτικής πλοήγησης σε έξυπνες συσκευές, η οποία περιλαμβάνει και αξιολόγηση των προτεινόμενων διαδρομών από τους χρήστες χρησιμοποιώντας αρχές πληθοπορισμού. Πιο συγκεκριμένα :

- (α) Μελετήθηκαν εκτενώς σύγχρονα γραφοκεντρικά δυναμικά μοντέλα αναπαράστασης μεγάλου όγκου δεδομένων ως προς την καταλληλότητα αναπαράστασης πινάκων δρομολογίων. Η μελέτη επιβεβαίωσε την καταλληλότητα του ρεαλιστικού χρονοεκτεταμένο μοντέλου.
- (β) Αναπτύχθηκαν δύο νέα γραφοκεντρικά μοντέλα αναπαράστασης πινάκων δρομολογίων, το *ανηγμένο χρονοεκτεταμένο μοντέλο* και το *δυναμικό μοντέλο πινάκων δρομολογίων* DTM, τα οποία είναι χωρο-αποδοτικότερα σε

σχέση με το ρεαλιστικό χρονοεκτεταμένο μοντέλο. Για τα νέα μοντέλα, αναπτύχθηκαν αποδοτικοί αλγόριθμοι απάντησης ερωτημάτων ΝΑ και Ε-ΑΜ, καθώς και αποδοτικοί αλγόριθμοι ενημέρωσης των γραφοκεντρικών μοντέλων σε περίπτωση καθυστερήσεων.

- (γ) Αξιολογήθηκαν πειραματικά τα νέα γραφοκεντρικά μοντέλα αναπαράστασης πινάκων δρομολογίων και οι νέοι αποδοτικοί αλγόριθμοι απάντησης ερωτημάτων και ενημέρωσης σε περίπτωση καθυστερήσεων, σε 14 σύνολα πραγματικών συγκοινωνιακών δεδομένων ΜΜΜ διαφόρων μεγεθών, τα οποία που περιλαμβάνουν τις μητροπολιτικές περιοχές του Βερολίνου, του Λονδίνου, της Ρώμης, της Αθήνας και της Μαδρίτης. Τα πειραματικά αποτελέσματα κατέδειξαν ότι οι αλγόριθμοι απάντησης ερωτημάτων του ανηγμένου χρονοεκτεταμένου μοντέλου υπερτερούν εκείνων του μοντέλου DTM, ενώ το αντίθετο συμβαίνει για τον αλγόριθμο ενημέρωσης σε περίπτωση καθυστερήσεων. Επίσης, τα πειραματικά αποτελέσματα κατέδειξαν ότι οι χρόνοι απόκρισης ερωτημάτων των νέων γραφοκεντρικών μοντέλων είναι συγκρίσιμοι, ή/και καλύτεροι, από/με αυτούς των γνωστών διανυσματοκεντρικών μοντέλων.
- (δ) Αναπτύχθηκε μια εφαρμογή κινητών συσκευών για ταξιδιωτική πλοήγηση σε αστικά περιβάλλοντα με χρήση ΜΜΜ, βασισμένη στα νέα γραφοκεντρικά μοντέλα, η οποία παρέχει την δυνατότητα στους χρήστες να αξιολογούν τις προτεινόμενες διαδρομές. Η εφαρμογή καταδεικνύει την πρακτική εφαρμοσιμότητα των προτεινόμενων μοντέλων και αλγορίθμων.

Abstract

An ever growing emphasis is put nowadays in developing personalized journey planning and renewable mobility services in smart cities. These services combine means of scheduled-based public transport and electric vehicles or bikes, using crowdsourcing techniques for collecting real-time traffic information and for assessing the recommended routes. The goal is to develop a system that will allow the fast, real-time computation of best routes.

The main challenges in developing such a system are both technological and algorithmic. The technological challenge concerns the collection, storage, management, and updating of a huge volume of transport data that are usually time-dependent, and the provision (through these data) of personalized renewable mobility services in smartphones. This challenge is typically confronted by creating a cloud infrastructure that on the one hand will support the storage, management, and updating of data, while on the other hand it will handle the necessary data feed to the smartphone applications for providing the users with the requested best routes.

The algorithmic challenge concerns the development of innovative algorithms for the *efficient* provision of journey planning services in smartphones, based on data they will receive from the cloud infrastructure. These services guarantee the computation of realistic and useful best routes, as well as the updating of the precomputed (route and timetable) information, in case of delays of scheduled public transport vehicles, so that the users can online update their routes to destination. The goal is to develop an algorithmic basis for supporting modern renewable mobility services, such as “mobility on demand” (where the next leg of a journey is decided in real-time) and “door-to-door” personalized mobility, in urban scheduled-based public transport environments. Scheduled-based public transport systems should not only compute in real-time end-user queries requesting best routes, but also to update the timetable information in case of delays.

The core algorithmic issues of mobility and journey planning (regarding the computation of optimal routes under certain criteria) in scheduled-based

public transport systems concern the efficient solution of the fundamental *earlier arrival* (EA) problem (compute a journey from station S to station T minimizing the overall traveling time required to complete the journey), the *minimum number of transfers* (MNT) problem (compute a journey from station S to station T minimizing the number of times a passenger is required to change vehicle), and the efficient updating of timetable information in case of vehicle delays. The EA and MNT problems have been extensively studied in the literature under two main approaches: the *array-based* modeling (where the timetable is represented as an array) and the *graph-based* modeling (where the timetable is represented as a graph). Experimental results have shown so far that the array-based approaches are faster in terms of query time than graph-based ones, as they are able to better exploit data locality and do not rely on priority queues. On the other hand, the array-based approaches have not been theoretically or experimentally studied as far as the efficient updating of timetable information, in case of delays, is concerned.

In this thesis, new graph-based models are being developed that solve efficiently the aforementioned fundamental algorithmic mobility problems in urban scheduled-based public transport systems, along with a mobile application (journey planner) running on Android-based smartphones that includes a service for the evaluation of the recommended routes by the users. In particular:

- (a) An extensive comparative evaluation was conducted on graph-based dynamic models that represent big data volumes regarding their suitability for representing timetable information. The study confirmed that the realistic time-expanded model is the most suitable for representing timetable information.
- (b) Two new graph-based models have been developed for representing timetable information, the *reduced time-expanded* model and the *dynamic timetable* model (DTM), both of which are more space-efficient with respect to the realistic time-expanded model. For both of the new models, new efficient algorithms were developed for fast answering of

EA and MNT queries, as well as for updating the timetable information representation in case of delays.

- (c) An extensive experimental evaluation was conducted with the new graph-based models and their associated query and update algorithms on a set of 14 real-world scheduled-based transportation systems, including the metropolitan areas of Berlin, Athens, London, Rome, and Madrid. The experimental results showed that the query algorithms of the reduced time-expanded model are superior to those of the DTM model, while the reverse is true regarding the update algorithms. In addition, the experimental study showed that the query algorithms of the new graph-based models compete favorably with those of the best array-based models.
- (d) A mobile, cloud-based, journey planner was developed whose core algorithmic engine builds upon the new graph-based models. The mobile application is accompanied by a service that allows the users to assess the recommended journeys. The journey planner demonstrates the practicality of the new graph-based models and their associated query and update algorithms.

Περιεχόμενα

1 Εισαγωγή	17
1.1 Εύρεση Βέλτιστων Διαδρομών σε Συγκοινωνιακά Συστήματα MMM	18
1.1.1 Τρέχοντα Συστήματα Πληροφορίας Δρομολογίων MMM	20
1.1.2 Βασικά Αλγοριθμικά Ζητήματα και Τρέχουσα Αντιμετώπιση	21
1.2 Στόχος Διδακτορικής Διατριβής	25
1.3 Συνεισφορά Διδακτορικής Διατριβής	26
1.4 Δομή	28
2 Βασικές Έννοιες και Μαθηματικό Υπόβαθρο	31
2.1 Γραφήματα	31
2.2 Συντομότερες Διαδρομές	33
2.3 Αλγόριθμος Dijkstra	34
2.3.1 Αποδοτική Υλοποίηση Αλγορίθμου Dijkstra	35
2.4 Ερωτήματα Συντομότερων Διαδρομών και Ευρετικές Μέθοδοι	38
2.4.1 Ευρετική Τεχνική A^*	39
2.4.2 Ευρετική Τεχνική ALT	40
3 Γραφοκεντρικά Μοντέλα	45
3.1 Βασικές Έννοιες Αναπαράστασης Πινάκων Δρομολογίων	45
3.2 Ρεαλιστικό Χρονοεκτεταμένο Μοντέλο	48
3.2.1 Μοντέλο	49
3.2.2 Αλγόριθμος Απάντησης Ερωτημάτων	51
3.2.3 Αλγόριθμος Ενημέρωσης Καθυστερήσεων	52
3.3 Άλλα Σύγχρονα Δυναμικά Γραφοκεντρικά Μοντέλα	54
3.3.1 Ορισμοί	56

3.3.2 Τύποι Ερωτημάτων	57
3.3.3 Κεντρικοποιημένα Μοντέλα	58
3.3.4 Αποκεντρικοποιημένα Μοντέλα	65
3.4 Αποτίμηση Μοντέλων	68
4 Νέα Γραφοκεντρικά Μοντέλα Αναπαράστασης Πληροφορίας Δρομολογίων	71
4.1 Ανηγμένο Ρεαλιστικό Χρονοεκτεταμένο Μοντέλο	72
4.1.1 Το Μοντέλο	72
4.1.2 Αλγόριθμος Απάντησης Ερωτημάτων	74
4.1.3 Αλγόριθμος Ενημέρωσης Καθυστερήσεων	76
4.2 Το Δυναμικό Μοντέλο Πίνακα Δρομολογίων	79
4.2.1 Το Μοντέλο	79
4.2.2 Αλγόριθμος Απάντησης Ερωτημάτων	81
4.2.3 Αλγόριθμος Ενημέρωσης Καθυστερήσεων	88
4.2.4 Επιτάχυνση Αλγορίθμου Απάντησης Ερωτημάτων	92
4.3 Σύγκριση των Χρονοεκτεταμένων Μοντέλων	94
4.4 Συνδυασμός Γραφοκεντρικών Μοντέλων με την Ευρετική Τεχνική ALT	95
5 Πειραματική Αξιολόγηση Νέων Μοντέλων	99
5.1 Δεδομένα Εισόδου και Παράμετροι	100
5.2 Ερωτήματα Πινάκων Δρομολογίων	103
5.3 Ενημέρωση Πινάκων Δρομολογίων	106
5.4 Σύγκριση με τον RAPTOR	108
6 Εφαρμογή Κινητών Συσκευών	111
6.1 Εισαγωγή	111
6.2 Το Περιβάλλον Ανάπτυξης του Ταξιδιωτικού Πλοηγού	114
6.2.1 Το Περιβάλλον Κινητών Συσκευών	114
6.2.2 Η Πλατφόρμα Organicity	115
6.3 Ο Ταξιδιωτικός Πλοηγός	120
6.3.1 Λειτουργικότητα Εφαρμογής	121
6.3.2 Υπηρεσία Αξιολόγησης Διαδρομών	126

7 Συμπεράσματα και Προοπτικές	129
7.1 Συμπεράσματα	129
7.2 Προοπτικές	130

Κατάλογος Σχημάτων

Σχήμα 2.1:	Αλγόριθμος Dijkstra	35
Σχήμα 2.2:	Αλγόριθμος Dijkstra με ουρά προτεραιότητας	37
Σχήμα 2.3:	Ορόσημα και τριγωνική ανισότητα	41
Σχήμα 3.1:	Το ρεαλιστικό χρονοεκτεταμένο μοντέλο (γράφημα)	50
Σχήμα 3.2:	Εξελισσόμενη Ακολουθία Γραφημάτων	56
Σχήμα 3.3:	Γραφήματα G_U και $G_\cap$ εξελισσόμενης ακολουθίας γρα- φημάτων	59
Σχήμα 3.4:	Δομές EdgeLog και EveLog εξελισσόμενης ακολουθίας γραφημάτων	63
Σχήμα 4.1:	Ανηγμένο χρονοεκτεταμένο μοντέλο (γράφημα)	73
Σχήμα 4.2:	Διαχείριση καθυστερήσεων στο ανηγμένο χρονοεκτεταμέ- νο μοντέλο	76
Σχήμα 4.3:	Το δυναμικό μοντέλο πίνακα δρομολογίων (DTM)	81
Σχήμα 4.4:	Διαχείριση καθυστερήσεων στο μοντέλο DTM	89
Σχήμα 6.1:	Αρχιτεκτονική της πλατφόρμας Organicity	116
Σχήμα 6.2:	Το πλαίσιο Πειραματισμού-ως-Υπηρεσία της πλατφόρ- μας Organicity	118
Σχήμα 6.3:	Ο ταξιδιωτικός πλοηγός.	120
Σχήμα 6.4:	Αρχική οθόνη ταξιδιωτικού πλοηγού και επιλογή πόλης .	122
Σχήμα 6.5:	Είσοδος στοιχείων για υπολογισμό διαδρομής	123
Σχήμα 6.6:	Ερώτημα και απεικόνιση βέλτιστης διαδρομής	124
Σχήμα 6.7:	Οδηγίες πλοήγησης	125
Σχήμα 6.8:	Επιλογή μέσου μεταφοράς	126
Σχήμα 6.9:	Υπηρεσία αξιολόγησης διαδρομών	127

Σχήμα 6.10: Αξιολόγηση διαδρομής	128
--	-----

Κατάλογος Πινάκων

Πίνακας 3.1: Σύνοψη Σύγχρονων Γραφοκεντρικών Μοντέλων	68
Πίνακας 5.1: Εξεταζόμενοι πίνακες δρομολογίων και αναπαράσταση στο μοντέλο TE-real	100
Πίνακας 5.2: Εξεταζόμενοι πίνακες δρομολογίων και αναπαράσταση στα μοντέλα TE-red και DTM	101
Πίνακας 5.3: Χαρακτηριστικά πινάκων δρομολογίων (1)	102
Πίνακας 5.4: Χαρακτηριστικά πινάκων δρομολογίων (2)	102
Πίνακας 5.5: Χρόνος και χώρος προεπεξεργασίας ευρετικής τεχνικής ALT	103
Πίνακας 5.6: Σύγκριση χρόνων μεταξύ αλγορίθμων απόκρισης ερω- τημάτων	104
Πίνακας 5.7: Σύγκριση μεταξύ αλγορίθμων ενημέρωσης πινάκων δρο- μολογίων	107
Πίνακας 5.8: Σύγκριση νέων γραφοκεντρικών μοντέλων και RAPTOR	109

Κεφάλαιο 1

Εισαγωγή

Τα τελευταία χρόνια δίνεται συνεχώς όλο και μεγαλύτερη έμφαση στις προσωποποιημένες υπηρεσίες ανανεωνόμενης κινητικότητας (personalized renewable mobility services) μέσα σε έξυπνες πόλεις. Οι υπηρεσίες αυτές συνδυάζουν Μέσα Μαζικής Μεταφοράς (MMM) και μικρά ηλεκτρικά αυτοκίνητα ή ποδήλατα, χρησιμοποιώντας τη λεγόμενη τεχνική του πληθοπορισμού (crowd-sourcing) για συλλογή δεδομένων κίνησης πραγματικού χρόνου και αξιολόγησης συνιστώμενων διαδρομών. Στόχος είναι η δημιουργία μιας γνωσιακής βάσης συγκοινωνιακών δεδομένων που θα επιτρέπει τη γρήγορη εύρεση βέλτιστων διαδρομών, θα προσαρμόζεται στις πραγματικές συνθήκες κίνησης και θα παρέχει ενεργειακά και περιβαλλοντικά αποδοτικές προσωποποιημένες υπηρεσίες ανανεωνόμενης κινητικότητας στους πολίτες.

Οι βασικότερες προκλήσεις στο σχεδιασμό ενός τέτοιου συστήματος είναι τεχνολογικές και αλγοριθμικές. Η τεχνολογική πρόκληση αφορά τη συλλογή, αποθήκευση, διαχείριση, και ενημέρωση ενός τεράστιου όγκου συγκοινωνιακών δεδομένων που είναι συνήθως χρονοεξαρτώμενα, και η παροχή (μέσω αυτών) προσωποποιημένων υπηρεσιών ανανεωνόμενης κινητικότητας σε έξυπνες φορητές συσκευές. Αυτή η πρόκληση αντιμετωπίζεται συνήθως με τη δημιουργία ενός περιβάλλοντος υπολογιστικού νέφους (cloud), το οποίο αφενός θα υποστηρίξει την αποθήκευση, διαχείριση, και ενημέρωση των δεδομένων και αφετέρου θα φροντίζει για την παροχή πληροφοριών στις εφαρμογές των έξυπνων φορητών συσκευών για εύρεση βέλτιστων διαδρομών.

Η αλγοριθμική πρόκληση αφορά στην ανάπτυξη καινοτόμων αλγορίθμων, οι οποίοι είναι κομβικής σημασίας (όπως θα εξηγηθεί παρακάτω) για την παροχή αποδοτικών προσωποποιημένων υπηρεσιών ανανεωνόμενης κινητικότητας σε έξυπνες φορητές συσκευές, με δεδομένα που θα λαμβάνουν από το υπολογιστικό νέφος. Οι υπηρεσίες αυτές εξασφαλίζουν την εύρεση ρεαλιστικών και χρήσιμων βέλτιστων διαδρομών, αλλά και την άμεση παροχή ενημερωμένων διαδρομών στους συνδεδεμένους χρήστες, σε περίπτωση καθυστερήσεων, για την επικαιροποίηση του αρχικού πλάνου τους.

Το κύριο ζητούμενο είναι η ανάπτυξη αλγοριθμικών προσεγγίσεων για την επαρκή κάλυψη των προαναφερθεισών αλγοριθμικών προκλήσεων, ιδιαίτερα σε αστικά συγκοινωνιακά περιβάλλοντα MMM. Τα συστήματα MMM, που λειτουργούν βάσει συγκεκριμένου πίνακα δρομολογίων, πρέπει όχι μόνο να ανταποκρίνονται σε πραγματικό χρόνο σε ερωτήματα χρηστών για εύρεση βέλτιστων διαδρομών, αλλά και να προβαίνουν σε ενημερώσεις των αποθηκευμένων συγκοινωνιακών πληροφοριών προκειμένου να ανταποκρίνονται (επίσης σε πραγματικό χρόνο) στις συχνές αλλαγές των προκαθορισμένων δρομολογίων λόγω καθυστερήσεων.

1.1 Εύρεση Βέλτιστων Διαδρομών σε Συγκοινωνιακά Συστήματα MMM

Η εύρεση βέλτιστων διαδρομών σε ένα δημόσιο συγκοινωνιακό σύστημα MMM (που αποτελείται δηλαδή από τρένα, λεωφορεία, μετρό, τραμ, κλπ) είναι ένα πρόβλημα το οποίο συναντά οποιοσδήποτε ταξιδεύει.

Δεδομένου ενός πίνακα δρομολογίων, που συσχετίζεται με ένα δημόσιο σύστημα μεταφορών, το πρόβλημα της εύρεσης βέλτιστης διαδρομής ή *ταξιδιωτικής πλοήγησης* (journey planning) αφορά στην αποδοτική απάντηση ερωτημάτων της μορφής: «Ποια είναι η καλύτερη διαδρομή από κάποιον σταθμό A σε κάποιον άλλο σταθμό B , υπό την προϋπόθεση ότι θέλω να αναχωρήσω από τον σταθμό A την χρονική στιγμή t ;».

Το βασικό ζήτημα είναι αφενός η οργάνωση της πληροφορίας ενός πίνακα δρομολογίων έτσι ώστε τα ερωτήματα εύρεσης βέλτιστων διαδρομών που θα

τεθούν από δυνητικούς ταξιδιώτες να απαντηθούν αποδοτικά, και αφετέρου η αποδοτική διαχείριση και ταχύτατη απάντηση ενός πολύ μεγάλου πλήθους τέτοιων ερωτημάτων.

Σε αντίθεση με την απλή του διατύπωση, το πρόβλημα εύρεσης βέλτιστων διαδρομών με χρήση μέσων μαζικής μεταφοράς κρύβει σημαντικές προκλήσεις και είναι πολύ περισσότερο περίπλοκο από το αντίστοιχο πρόβλημα στα οδικά δίκτυα. Τα συστήματα μεταφορών βασισμένα σε πίνακες δρομολογίων απαιτούν, λόγω της χρονοεξαρτώμενης φύσης τους, μοντελοποίηση με περισσότερο πολύπλοκες παραδοχές προκειμένου να παράγουν λογικά και χρήσιμα αποτελέσματα, ιδίως όταν πρέπει να ληφθούν υπόψη και οι χρόνοι μετεπιβίβασης από ένα όχημα σε ένα άλλο σε κάποιον σταθμό.

Για το λόγο αυτό, πολλοί δημόσιοι συγκοινωνιακοί οργανισμοί και εταιρείες έχουν επενδύσει σημαντικά στην ανάπτυξη λογισμικού, που ονομάζεται *ταξιδιωτικός πλοηγός* (journey planner), για την παροχή μέσω εφαρμογών διαδικτύου ή/και κινητών συσκευών αποδοτικών απαντήσεων (σε πραγματικό χρόνο) σε ερωτήματα εύρεσης βέλτιστων διαδρομών με βάση διάφορες μετρικές.

Η αποδοτική επίλυση του παραπάνω προβλήματος είναι ζωτικής σημασίας στους δημόσιους συγκοινωνιακούς οργανισμούς (π.χ. στους Ευρωπαϊκούς σιδηρόδρομους), όπου στα σύγχρονα περιβάλλοντα των έξυπνων πόλεων ένας κεντρικός εξυπηρετητής (που συνήθως βρίσκεται σε μια δομή υπολογιστικού νέφους) πρέπει να απαντά, σε πραγματικό χρόνο, σε ερωτήματα δρομολογίων που τίθενται από τους ταξιδιώτες είτε μέσω τερματικών στους σταθμούς έκδοσης εισιτηρίων, είτε μέσω διαδικτυακών εφαρμογών, είτε μέσω εφαρμογών κινητών συσκευών. Ο εξυπηρετητής αυτός, θα πρέπει να μπορεί να ανταποκριθεί ταχύτατα σε ένα τεράστιο αριθμό ερωτημάτων ταξιδιωτών, οι οποίοι αναζητούν τις βέλτιστες διαδρομές για τους προορισμούς τους. Για παράδειγμα, ο κεντρικός εξυπηρετητής των Γερμανικών σιδηροδρόμων δέχεται κατά μέσο όρο 420 τέτοια ερωτήματα ανά δευτερόλεπτο σε ώρες αιχμής.

Ο κύριος στόχος σε όλες αυτές τις εφαρμογές είναι η μείωση του μέσου χρόνου ανταπόκρισης του εξυπηρετητή σε ένα ερώτημα.

1.1.1 Τρέχοντα Συστήματα Πληροφορίας Δρομολογίων MMM

Το πρώτο ηλεκτρονικό σύστημα πινάκων δρομολογίων εγκαταστάθηκε στα τέλη της δεκαετίας του 1980. Τα υπάρχοντα συστήματα είναι το HAFAS [50], το οποίο χρησιμοποιείται από τις περισσότερες Ευρωπαϊκές σιδηροδρομικές εταιρείες, και το EFA [36], το οποίο κυρίως χρησιμοποιείται για τη καταγραφή της κίνησης σε μικρότερες περιοχές της Ευρώπης.

Τα αποτελέσματα που προκύπτουν από τα παραπάνω συστήματα είναι στην πλειονότητα των περιπτώσεων ικανοποιητικά. Ωστόσο, υπάρχουν συχνά περιπτώσεις, όπου οι προτεινόμενες διαδρομές δεν είναι βέλτιστες σύμφωνα με το δοθέν κριτήριο βελτιστοποίησης. Ο βασικός λόγος που προκύπτουν μη βέλτιστες διαδρομές είναι ότι οι αλγόριθμοι, που στηρίζονται τα συστήματα αυτά, χρησιμοποιούν συνήθως συγκεκριμένες ευρετικές μεθόδους για να μειώσουν την περιοχή αναζήτησης (άρα και το χρόνο αναζήτησης), οι οποίες δίνουν καλές αλλά όχι πάντοτε βέλτιστες λύσεις. Αυτές οι ευρετικές μέθοδοι συνήθως εξελίσσονται σε δύο φάσεις.

Η πρώτη ευρετική μέθοδος περιλαμβάνεται στο σύστημα TRAINS [83] και χρησιμοποιήθηκε στους Γερμανικούς σιδηρόδρομους. Βασίζεται σε ένα γράφημα όπου οι κορυφές αναπαριστούν πόλεις. Το δίκτυο διακρίνεται σε δύο επίπεδα, το στατικό επίπεδο όπου οι ακμές αναπαριστούν αποστάσεις, και το δυναμικό επίπεδο όπου οι ακμές περιέχουν πληροφορίες για τους χρόνους αναχώρησης και άφιξης των τρένων. Ο αλγόριθμος χρησιμοποιεί το στατικό επίπεδο για να απομονώσει την ενδιαφέρουσα περιοχή του δικτύου, χωρίς να λαμβάνει υπόψιν καμία πληροφορία σχετικά με το χρόνο.

Η δεύτερη ευρετική μέθοδος περιλαμβάνεται στο σύστημα ARIADNE [10] και αποτέλεσε τον πρόγονο του συστήματος HAFAS [50]. Όπως στο σύστημα TRAINS, ο αλγόριθμος αποτελείται από δύο διαφορετικά δίκτυα: το στατικό και το δυναμικό. Το ARIADNE λειτουργεί σε δύο φάσεις. Στην πρώτη φάση, γίνεται η εύρεση εφικτών διαδρομών στο στατικό δίκτυο με την εφαρμογή του αλγόριθμου Dijkstra [23]. Το αποτέλεσμα της πρώτης φάσης είναι ένα υπογράφημα του αρχικού δικτύου το οποίο εισάγεται στη δεύτερη φάση. Όπως και στην προηγούμενη μέθοδο σε αυτό το βήμα μπορεί να χαθεί η βέλτιστη λύση. Στη δεύτερη φάση, οι εφικτές διαδρομές από το υπογράφημα που προέκυψε

κατά τη πρώτη φάση, κατατάσσονται με βάση μετρικές όπως ο χρόνος ταξιδιού, η ταχύτητα των τρένων, η απευθείας ανταπόκριση, και άλλες.

Τα τελευταία χρόνια έχει αναπτυχθεί εκτενώς και η εφαρμογή Google transit [48], η οποία προσφέρει βέλτιστες διαδρομές με διάφορα κριτήρια και επιλογή μέσων μαζικής μεταφοράς. Η εφαρμογή βασίζεται στην αξιοπιστία των δεδομένων που παρέχουν διάφοροι συγκοινωνιακοί ή ιδιωτικοί οργανισμοί.

Όλα τα παραπάνω συστήματα προσφέρουν δυνατότητα επικαιροποίησης των δεδομένων σε περίπτωση καθυστερήσεων, χωρίς να προσφέρουν πάντοτε τη δυνατότητα ενημέρωσης της αναπαράστασης του πίνακα δρομολογίων και την παροχή εναλλακτικής βέλτιστης διαδρομής που να λαμβάνει υπόψη τις καθυστερήσεις.

1.1.2 Βασικά Αλγοριθμικά Ζητήματα και Τρέχουσα Αντιμετώπιση

Ανάλογα με τις μετρικές που χρησιμοποιούνται και τις παραδοχές μοντελοποίησης, το πρόβλημα της ταξιδιωτικής πλοήγησης μπορεί να εξειδικευθεί σε μια πληθώρα προβλημάτων βελτιστοποίησης. Ένα πρώτο σύνολο προβλημάτων ορίζεται λαμβάνοντας υπόψιν διαφορετικές αντικειμενικές συναρτήσεις βελτιστοποίησης. Αν για παράδειγμα, η καλύτερη (*βέλτιστη*) διαδρομή είναι αυτή που ελαχιστοποιεί τον συνολικό χρόνο ταξιδιού, τότε το πρόβλημα ονομάζεται *πρόβλημα νωρίτερης άφιξης*. Μια άλλη, απλούστερη και λιγότερο γνωστή παραλλαγή του προβλήματος είναι η εύρεση βέλτιστης διαδρομής που ελαχιστοποιεί τον αριθμό των μετεπιβιβάσεων οι οποίες απαιτούνται από έναν επιβάτη προκειμένου να μετακινηθεί από ένα όχημα σε ένα άλλο, κατά τη διάρκεια του ταξιδιού του. Αυτή η εκδοχή του προβλήματος ονομάζεται *πρόβλημα ελάχιστου αριθμού μετεπιβιβάσεων*. Μερικές φορές, αυτά τα δύο κριτήρια βελτιστοποίησης μπορούν να συνδυαστούν ορίζοντας το *δικριτιριακό πρόβλημα βελτιστοποίησης*. Μπορούν επίσης να συνδυαστούν και με άλλα κριτήρια (όπως για παράδειγμα το χρηματικό κόστος), στην οποία περίπτωση προκύπτει ένα *πολυκριτιριακό πρόβλημα βελτιστοποίησης*.

Περαιτέρω εξειδικεύσεις μπορούν να προκύψουν ανάλογα με το επίπεδο της αφαιρετικής διαδικασίας κατά την μοντελοποίηση του προβλήματος που πρέ-

πει να επιλυθεί. Αν για παράδειγμα, η μοντελοποίηση και η βελτιστοποίηση πειλαμβάνει και τον χρόνο που απαιτείται από έναν επιβάτη για να μετακινηθεί από ένα όχημα σε ένα άλλο μέσα στον ίδιο σταθμό — δηλαδή, τον *χρόνο μεταπιβίδωσης* — τότε το πρόβλημα αναφέρεται ως *ρεαλιστικό* [72], ενώ όταν δεν λαμβάνεται υπόψιν ο χρόνος μεταπιβίδωσης το πρόβλημα αναφέρεται ως *ιδεατό* ή *βασικό*. Η παρούσα διδακτορική διατριβή, εστιάζεται μόνο στο ρεαλιστικό πρόβλημα ταξιδιωτικής πλοήγησης.

Επιπλέον, αν ο χρόνος αναχώρησης ανήκει σε ένα συγκεκριμένο χρονικό διάστημα και δεν είναι μια μεμονωμένη τιμή, τότε το πρόβλημα ονομάζεται *εύρεση νωρίτερης άφιξης με αναχώρηση εντός ενός χρονικού διαστήματος* (profile or range query problem). Η εργασία [6] παρέχει μια πρόσφατη και περιεκτική ερευνητική επισκόπηση αλγοριθμικών μεθόδων ταξιδιωτικής πλοήγησης.

Η μεγάλη ποικιλία μοντέλων που προτείνονται στη βιβλιογραφία για την επίλυση των προαναφερθέντων παραλλαγών του προβλήματος ταξιδιωτικής πλοήγησης μπορεί να ταξινομηθεί σε δύο μεγάλες κατηγορίες: σε εκείνες που αναπαριστούν τους πίνακες δρομολογίων ως *διανύσματα* και σε εκείνες που τους αναπαριστούν ως *γραφήματα*.

Δύο από τα πιο επιτυχημένα (και αποτελεσματικά) παραδείγματα των *διανυσματοκεντρικών μοντέλων* είναι ο αλγόριθμος Connection Scan (CSA) [33] και ο αλγόριθμος Round-based Public Transit Optimized Router (RAPTOR) [28, 29].

Στον CSA όλες οι *στοιχειώδεις συνδέσεις* ενός πίνακα δρομολογίων αποθηκεύονται σε ένα μοναδικό πίνακα ο οποίος σαρώνεται μόνο μία φορά ανά ερώτημα, για την επίλυση του προβλήματος νωρίτερης άφιξης. Αυτό προϋποθέτει ότι ο πίνακας δρομολογίων είναι ακυκλικός (δηλαδή απεριοδικός). Στον RAPTOR ο πίνακας δρομολογίων αποθηκεύεται ως ένα σύνολο διανυσμάτων ταξιδιών και διαδρομών που χρησιμοποιούνται από έναν αλγόριθμο δυναμικού προγραμματισμού για την επίλυση του δικριτιριακού προβλήματος. Πρόσφατα, στις εργασίες [25] και [86], προτάθηκαν μερικές ταχύτερες προσεγγίσεις όσον αφορά τον χρόνο απόκρισης ερωτημάτων σε σχέση με τις δύο προαναφερθείσες μεθόδους.

Τα *γραφοκεντρικά μοντέλα* αποθηκεύουν τον πίνακα δρομολογίων σε ένα κατάλληλο γράφημα και εκτελούν γνωστές παραλλαγές του αλγορίθμου συ-

ντομότερων διαδρομών Dijkstra [23] για τον υπολογισμό των βέλτιστων διαδρομών. Υπάρχουν δύο βασικές προσεγγίσεις στην κατηγορία των γραφοκεντρικών μοντέλων: το *χρονοεκτεταμένο* και το *χρονοεξαρτώμενο* [72]. Το πρώτο αναπαριστά ρητά το κάθε χρονικό συμβάν (αναχώρηση ή άφιξη) του πίνακα δρομολογίων με μία κορυφή στο αντίστοιχο γράφημα. Οι ακμές αντιπροσωπεύουν στοιχειώδεις συνδέσεις μεταξύ δύο χρονικών συμβάντων ή αναμονές εντός των σταθμών και το κόστος τους αντιστοιχεί συνήθως στη χρονική διαφορά μεταξύ των αντίστοιχων χρονικών συμβάντων. Το δεύτερο μοντέλο αναπαριστά κάθε σταθμό με μία κορυφή, ενώ υπάρχει ακμή μεταξύ δύο κόμβων αν υπάρχει τουλάχιστον μία στοιχειώδης σύνδεση μεταξύ των δύο σταθμών που αντιπροσωπεύονται από τις αντίστοιχες κορυφές. Το κόστος μιας ακμής είναι εξαρτώμενο από το χρόνο, δηλαδή είναι μια συνάρτηση που εξαρτάται από τον χρόνο κατά τον οποίο μία συγκεκριμένη ακμή σαρώνεται κατά τη διάρκεια της αναζήτησης της συντομότερης διαδρομής. Το χρονοεκτεταμένο μοντέλο παράγει ένα γράφημα με μεγαλύτερο αριθμό κορυφών και ακμών σε σχέση με το χρονοεξαρτώμενο μοντέλο και επομένως έχει και μεγαλύτερους χρόνους απόκρισης ερωτημάτων. Μια παραλλαγή του χρονοεκτεταμένου μοντέλου που έχει μικρότερο αριθμό κορυφών και ακμών, γνωστό ως *ρεαλιστικό χρονοεκτεταμένο μοντέλο*, έχει παρουσιαστεί στην εργασία [72]. Τα γραφοκεντρικά μοντέλα μπορούν γενικά εύκολα να προσαρμοστούν για την επίλυση των πολυκριτηριακών προβλημάτων, όπως και ο RAPTOR [6].

Τα πειραματικά αποτελέσματα έχουν δείξει μέχρι στιγμής ότι οι διανυσματοκεντρικές προσεγγίσεις είναι γρηγορότερες όσον αφορά το χρόνο απάντησης των ερωτημάτων από τις γραφοκεντρικές [6, 29, 33], καθώς είναι σε θέση να αξιοποιήσουν καλύτερα τα δεδομένα τοπικότητας και να μην στηρίζονται σε ουρές προτεραιότητας. Από την άλλη πλευρά, οι διανυσματοκεντρικές προσεγγίσεις (π.χ. [33, 25]) έχουν δοκιμαστεί σε περιπτώσεις πόλεων μητροπολιτικού μεγέθους ή/και εθνικά δίκτυα MMM που καλύπτουν συνήθως μόνο τη χρονική περίοδο μιας ημέρας (και αντιστοιχούν σε απεριόδικους πίνακες δρομολογίων), ενώ οι γραφοκεντρικές προσεγγίσεις μπορούν να χρησιμοποιηθούν σε περιοδικούς πίνακες δρομολογίων (δηλαδή, καλύπτουν χρονικό διάστημα μεγαλύτερο από μία ημέρα) και επομένως οι δεύτερες είναι καταλληλότερες για τη διαχείριση ερωτημάτων σε δίκτυα ευρείας κλίμακας. Επιπλέον, τα τελευταία χρόνια μια

μεγάλη ερευνητική προσπάθεια έχει επικεντρωθεί στην ανάπτυξη των ονομαζόμενων *τεχνικών επιτάχυνσης*, οι οποίες επιταχύνουν ευρετικά τον αλγόριθμο του Dijkstra για την εύρεση συντομότερων διαδρομών (δείτε π.χ. τις εργασίες [6, 8]). Αυτές οι τεχνικές επικεντρώνονται κυρίως στην εύρεση βέλτιστων διαδρομών σε *οδικά δίκτυα*, όπου εμφανίζουν τεράστια επιτάχυνση σε σχέση με τον βασικό αλγόριθμο του Dijkstra. Επομένως, μια πολλά υποσχόμενη προσέγγιση θα μπορούσε να είναι η προσαρμογή των τεχνικών επιτάχυνσης που σχεδιάστηκαν για τα οδικά δίκτυα σε γραφήματα πινάκων δρομολογίων (δείτε π.χ. τις εργασίες [9, 27]). Σε αυτή την κατεύθυνση, έχει παρουσιαστεί μια τροποποίηση του ρεαλιστικού χρονεκτεταμένου μοντέλου και έχει αποδειχθεί ότι εναρμονίζεται πολύ καλά με αρκετές γνωστές τεχνικές επιτάχυνσης [27].

Ένας άλλος περιορισμός των γραφοκεντρικών μοντέλων αφορά στο ότι δεν φαίνονται κατάλληλα να ενσωματώσουν δυναμικές αλλαγές στον πίνακα δρομολογίων. Στην πραγματικότητα, αν η χρονική διάρκεια κάποιας σύνδεσης αλλάξει (π.χ. λόγω καθυστέρησης ενός τρένου), τότε το γράφημα μπορεί να μην αντιπροσωπεύει σωστά τον τροποποιημένο πίνακα δρομολογίων και επομένως η διαδρομή που υπολογίζει ο αλγόριθμος πιθανόν να είναι μη βέλτιστη, ή ακόμη μπορεί να είναι και μη εφικτή. Για παράδειγμα, μια μελέτη που πραγματοποιήθηκε στο δημόσιο συγκοινωνιακό δίκτυο της Ρώμης, έδειξε ότι η εκμετάλλευση του δημοσιευμένου πίνακα δρομολογίων δεν οδηγεί σε βέλτιστες ή σχεδόν βέλτιστες διαδρομές [38]. Η ενημέρωση των γραφημάτων σύμφωνα με τις τροποποιήσεις των πινάκων δρομολογίων είναι χρονοβόρα και σε πολλές περιπτώσεις απαιτεί *τοπολογικές αλλαγές* του γραφήματος (δηλαδή προσθήκες και διαγραφές ακμών ή κορυφών) [26].

Επιπλέον, αν και έχουν αναπτυχθεί τεχνικές επιτάχυνσης για τα οδικά δίκτυα, οι οποίες επιτρέπουν τη διαχείριση δυναμικών ενημερώσεων [14, 21, 22, 30, 31, 34, 75, 84], οι τεχνικές αυτές δεν μπορούν να εφαρμοστούν στην περίπτωση πιθανών αλλαγών στον πίνακα δρομολογίων. Αυτό οφείλεται στο γεγονός ότι οι περισσότερες από αυτές τις τεχνικές βασίζονται στην προεπεξεργασία πρόσθετων πληροφοριών που αργότερα εκμεταλλεύονται για την απόκριση ερωτημάτων. Όταν συμβαίνει μια τροποποίηση στον πίνακα δρομολογίων, οι προεπεξεργασμένες πληροφορίες δεν είναι πλέον αξιόπιστες και πρέπει να επαναυπολογισθούν εξ' αρχής, απαιτώντας συνήθως μεγάλο υπολογιστικό χρόνο.

Σημαντική επιβάρυνση παρέχουν και πάλι οι τοπολογικές αλλαγές στο γράφημα. Η δυναμική συμπεριφορά των μοτίβων μεταφοράς (Transfer Patterns), που αποτελεί μια τεχνική επιτάχυνσης ειδικά σχεδιασμένη για δημόσια συστήματα μεταφορών [5], μελετήθηκε στην εργασία [7]. Εκεί καταδεικνύεται ότι χωρίς την εκτέλεση της εξ' αρχής προεπεξεργασίας, η τεχνική αυτή δίνει βέλτιστα αποτελέσματα για τη συντριπτική πλειοψηφία — αλλά *όχι για το σύνολο* — των ερωτημάτων. Ένα πρόβλημα άμεσης απόκρισης στο οποίο οι καθυστερήσεις αναφέρονται συνεχώς στον ταξιδιωτικό πλοηγό έχει μελετηθεί στην εργασία [66].

Όσον αφορά τα διανυσματοκεντρικά μοντέλα, οι αλγόριθμοι CSA και RPTOR είναι κατάλληλοι να διαχειρίζονται δυναμικές αλλαγές του πίνακα δρομολογίων δεδομένου ότι δεν βασίζονται στην προεπεξεργασία. Δεν υπάρχουν όμως στη βιβλιογραφία πειραματικές μελέτες για την πρακτική απόδοσή τους σε δυναμικές αλλαγές. Όσον αφορά τις ταχύτατες χρονικά προσεγγίσεις απάντησης ερωτημάτων που παρουσιάζονται στις εργασίες [25] και [86], η πρώτη δεν ενδείκνυται για δυναμικά σενάρια λόγω του τεράστιου χρόνου προεπεξεργασίας της, ενώ η δεύτερη φαίνεται καταλληλότερη, παρόλο που χρειάζεται περίπου 6 λεπτά προεπεξεργασίας σε ένα στιγμιότυπο που μοντελοποιεί το σύστημα δημόσιων συγκοινωνιών του Λονδίνου για ένα σταθερό χρονικό διάστημα (που περιλαμβάνει περίπου 5 εκατομμύρια συνδέσεις).

1.2 Στόχος Διδακτορικής Διατριβής

Στόχος της διδακτορικής αυτής διατριβής είναι η αποδοτική επίλυση θεμελιωδών αλγοριθμικών ζητημάτων για την παροχή υπηρεσιών ανανεωνόμενης και προσωποποιημένης κινητικότητας σε έξυπνες πόλεις. Σκοπός είναι η δημιουργία μιας αλγοριθμικής βάσης για την υποστήριξη υπηρεσιών ανανεωνόμενης κινητικότητας κατ' απαίτηση (*mobility on demand*), όπου το επόμενο τμήμα ενός ταξιδιού αποφασίζεται σε πραγματικό χρόνο, και υπηρεσιών προσωποποιημένης ανανεωνόμενης κινητικότητας για πλοήγηση «από-πόρτα-σε-πόρτα».

Στο αυτό το πλαίσιο, τα κεντρικά αλγοριθμικά ζητήματα των προβλημά-

των κινητικότητας με χρήση MMM σε έξυπνα αστικά περιβάλλοντα, τα οποία μελετώνται σε αυτή τη διατριβή, περιλαμβάνουν:

- (α) Την αποδοτική μοντελοποίηση (αναπαράσταση και διαχείριση) της πληροφορίας πινάκων δρομολογίων σε συγκοινωνιακά συστήματα MMM, που εμπεριέχει την πρόκληση της υψηλής πολυπλοκότητας και του μεγάλου μεγέθους τους.
- (β) Την ανάπτυξη νέων γραφοκεντρικών μοντέλων με απόδοση συγκρίσιμη με εκείνη των υπάρχοντων διανυσματοκεντρικών μοντέλων, τουλάχιστον για την απάντηση των θεμελιωδών προβλημάτων νωρίτερης άφιξης (NA) και ελάχιστου αριθμού μετεπιβιβάσεων (EAM).
- (γ) Την ανάπτυξη αποδοτικών αλγορίθμων ενημέρωσης, σε πραγματικό χρόνο, της πληροφορίας των γραφοκεντρικών μοντέλων αναπαράστασης πινάκων δρομολογίων σε περίπτωση καθυστέρησης οχημάτων, έτσι ώστε να μπορεί να επαναυπολογισθεί η νέα βέλτιστη διαδρομή (λαμβάνοντας υπόψη τις καθυστερήσεις).

Επιπλέον στόχος της διδακτορικής αυτής διατριβής είναι η διερεύνηση της εφαρμοσιμότητας και πρακτικής αξίας των προτεινόμενων αλγοριθμικών λύσεων σε πραγματικά δεδομένα συγκοινωνιακών συστημάτων MMM.

1.3 Συνεισφορά Διδακτορικής Διατριβής

Στο πλαίσιο της Ενότητας 1.2, η συνεισφορά της διδακτορικής διατριβής αφορά στην ανάπτυξη αλγοριθμικών μεθόδων για θεμελιώδη ζητήματα κινητικότητας πολιτών σε έξυπνες πόλεις με χρήση Μέσων Μαζικής Μεταφοράς (MMM), καθώς και στη δημιουργία εφαρμογής κινητικότητας σε έξυπνες συσκευές η οποία περιλαμβάνει και την αξιολόγηση των προτεινόμενων διαδρομών από τους χρήστες, χρησιμοποιώντας αρχές πληθοπορισμού.

Προτείνονται νέες αλγοριθμικές λύσεις για τα θεμελιώδη προβλήματα της νωρίτερης άφιξης (NA - μετακίνηση ενός επιβάτη από ένα σταθμό S σε ένα σταθμό T , στον ελάχιστο δυνατό χρόνο), του ελάχιστου αριθμού μετεπιβιβάσεων (EAM - μετακίνηση ενός επιβάτη από ένα σταθμό S σε ένα σταθμό T ,

με τον ελάχιστο αριθμό μετεπιβιβάσεων), και της δυναμικής ενημέρωσης της πληροφορίας των πινάκων δρομολογίων σε περίπτωση καθυστερήσεων.

Πιο συγκεκριμένα, τα αποτελέσματα της διδακτορικής διατριβής περιλαμβάνουν:

- Εκτενή μελέτη σύγχρονων γραφοκεντρικών δυναμικών μοντέλων αναπαράστασης μεγάλου όγκου δεδομένων. Μελετήθηκαν τόσο κεντριοποιημένα όσο και αποκεντρωμένα μοντέλα, αναλύθηκαν τα χαρακτηριστικά τους και η καταλληλότητα αποδοτικής επίλυσης των παραπάνω αλγοριθμικών προβλημάτων. Το ρεαλιστικό χρονοεκτεταμένο μοντέλο (TE) αποδείχτηκε ότι είναι το καταλληλότερο για την αναπαράσταση προβλημάτων που βασίζονται σε πίνακες δρομολογίων.
- Ανάπτυξη ενός νέου γραφοκεντρικού μοντέλου (Ανηγμένο Χρονοεκτεταμένο Μοντέλο, TE-red) αναπαράστασης πινάκων δρομολογίων, το οποίο μειώνει κατά $1/3$ το μέγεθος της δομής του (αρχικού) μοντέλου TE. Επίσης στο μοντέλο TE-red αναπτύχθηκαν νέοι αποδοτικοί αλγόριθμοι για την επίλυση των προβλημάτων NA και EAM, καθώς και ένας νέος αλγόριθμος ενημέρωσης της δομής αναπαράστασης πινάκων δρομολογίων στην περίπτωση καθυστερήσεων.
- Ανάπτυξη ενός νέου (δεύτερου) γραφοκεντρικού μοντέλου (Δυναμικό Μοντέλο Πινάκων Δρομολογίων, DTM) αναπαράστασης πινάκων δρομολογίων που επιτυγχάνει περαιτέρω μείωση του μεγέθους του γραφήματος στο $1/3$ του αρχικού ρεαλιστικού χρονοεκτεταμένου μοντέλου TE. Στο μοντέλο DTM αναπτύχθηκαν νέοι αποδοτικοί αλγόριθμοι για την επίλυση των προβλημάτων NA και EAM, καθώς και νέος αλγόριθμος ενημέρωσης του μοντέλου DTM στην περίπτωση καθυστερήσεων, ο οποίος διαχειρίζεται τις καθυστερήσεις σε γραμμικό χρόνο χωρίς καμία αλλαγή στην τοπολογία του γραφήματος.
- Εκτενή πειραματική αξιολόγηση των παραπάνω νέων μοντέλων και αλγορίθμων σε 14 σύνολα πραγματικών συγκοινωνιακών δεδομένων MMM διαφόρων μεγεθών, που περιλαμβάνουν τις μητροπολιτικές περιοχές του Βερολίνου, του Λονδίνου, της Ρώμης, της Αθήνας και της Μαδρίτης. Τα

πειραματικά αποτελέσματα έδειξαν ότι στο μοντέλο TE-red οι αλγόριθμοι επίλυσης NA και EAM είναι αποδοτικότεροι σε σχέση με τους αντίστοιχους του μοντέλου DTM, με χρόνους απόκρισης ερωτημάτων σε λίγα msec ακόμη και σε πίνακες δρομολογίων με 20,000 σταθμούς και 48,000,000 συνδέσεις MMM. Το αντίστροφο συμβαίνει στην περίπτωση ενημέρωσης καθυστερήσεων της δομής πινάκων δρομολογίων, δηλαδή, ο αλγόριθμος ενημέρωσης του μοντέλου DTM εκτελείται σε χρόνο < 0.3 msec, ενώ εκείνος του μοντέλου TE-red χρειάζεται περίπου διπλάσιο χρόνο (0.6 msec). Τέλος, η πειραματική αξιολόγηση κατέδειξε ότι οι χρόνοι απόκρισης ερωτημάτων των νέων γραφοκεντρικών μοντέλων είναι συγκρίσιμοι, ή/και καλύτεροι, από/με αυτούς των γνωστών διανυσματοκεντρικών μοντέλων.

- Ανάπτυξη εφαρμογής κινητών συσκευών για εύρεση βέλτιστων διαδρομών με χρήση MMM σε περιβάλλον Android, η οποία παρέχει την δυνατότητα στους χρήστες να αξιολογούν τις διαδρομές. Η αξιολόγηση πραγματοποιήθηκε με την ανάπτυξη μιας νέας υπηρεσίας που χρησιμοποιεί την πλατφόρμα διαδικτύου αντικειμένων (IoT) Organicity. Η ανάπτυξη της εφαρμογής κινητών συσκευών, σε συνδυασμό με την προαναφερθείσα εκτενή πειραματική αξιολόγηση, καταδεικνύει την πρακτική εφαρμοσιμότητα των προτεινόμενων μοντέλων και αλγορίθμων.

1.4 Δομή

Το υπόλοιπο της διδακτορικής διατριβής αποτελείται από έξι κεφάλαια των οποίων η οργάνωση έχει ως εξής.

Στο Κεφάλαιο 2 παρουσιάζονται βασικοί ορισμοί, συμβολισμοί και εισαγωγικά στοιχεία τα οποία είναι απαραίτητα για την ευκολότερη κατανόηση της διδακτορικής διατριβής. Επίσης, παρουσιάζεται το βασικό πρόβλημα εύρεσης συντομότερης διαδρομής από μία αρχική σε μία τελική κορυφή, ο βασικός αλγόριθμος του Dijkstra που το επιλύει, καθώς και ευρετικές τεχνικές που επιτυγχίνουν την εκτέλεση του αλγορίθμου στην πράξη.

Στο Κεφάλαιο 3 ορίζονται φορμαλιστικά τα προβλήματα που μελετά η διδα-

κτορική διατριβή και παρουσιάζεται το βασικό γραφοκεντρικό μοντέλο αναπαράστασης πληροφορίας πινάκων δρομολογίων. Παρουσιάζονται επίσης άλλα σύγχρονα δυναμικά γραφοκεντρικά μοντέλα αναπαράστασης μεγάλου όγκου πληροφορίας, και γίνεται μια συγκριτική αξιολόγηση όλων των μοντέλων. Το κείμενο του κεφαλαίου βασίζεται στη δημοσίευση [56].

Στο Κεφάλαιο 4 παρουσιάζονται δύο νέα γραφοκεντρικά μοντέλα αναπαράστασης πληροφορίας πινάκων δρομολογίων που σχεδιάστηκαν και αναπτύχθηκαν στα πλαίσια της παρούσας διδακτορικής διατριβής. Παρουσιάζεται το ανηγμένο χρονοκεκτεταμένο μοντέλο που είναι μία απλούστευση του αυθεντικού ρεαλιστικού μοντέλου. Στη συνέχεια περιγράφεται το νέο δυναμικό μοντέλο αναπαράστασης πινάκων δρομολογίων, το οποίο διαφοροποιείται από τα δύο προηγούμενα. Επιπλέον, και για τα δύο νέα μοντέλα δίνονται οι αλγόριθμοι επίλυσης των βασικών προβλημάτων συντομότερης άφιξης και ελάχιστου αριθμού μετεπιβιβάσεων, εμπλουτισμένων με τεχνικές επιτάχυνσης, καθώς και αλγόριθμοι για την αντιμετώπιση καθυστερήσεων. Το κείμενο του κεφαλαίου βασίζεται στις δημοσιεύσεις [19, 20, 32].

Στο Κεφάλαιο 5 παρουσιάζεται εκτενής πειραματική αξιολόγηση της απόδοσης όλων των αλγορίθμων των νέων γραφοκεντρικών μοντέλων που παρουσιάστηκαν στο Κεφάλαιο 4. Επιπλέον, γίνεται και σύγκριση με τον αλγόριθμο του πιο γνωστού διανυσματοκεντρικού μοντέλου RAPTOR. Το κείμενο του κεφαλαίου βασίζεται στις δημοσιεύσεις [19, 20].

Στο Κεφάλαιο 6 παρουσιάζεται ένας ψηφιακός ταξιδιωτικός πλοηγός μέσων μαζικής μεταφοράς που βασίζεται σε υποδομή υπολογιστικού νέφους και λειτουργεί σε έξυπνες κινητές συσκευές. Ο ψηφιακός ταξιδιωτικός πλοηγός περιέχει μία αποδοτική αλγοριθμική μηχανή δρομολόγησης η οποία βασίζεται στους αποδοτικούς αλγόριθμους εύρεσης βέλτιστων διαδρομών του Κεφαλαίου 4. Η εφαρμογή κινητών συσκευών παρέχει μια υπηρεσία που επιτρέπει στους χρήστες να αξιολογήσουν τις προτεινόμενες διαδρομές που προσφέρει η εφαρμογή. Το κείμενο του κεφαλαίου βασίζεται στις δημοσιεύσεις [44, 45].

Η διδακτορική διατριβή ολοκληρώνεται με το Κεφάλαιο 7, στο οποίο παρουσιάζονται τα συμπεράσματα και οι προοπτικές της.

Κεφάλαιο 2

Βασικές Έννοιες και Μαθηματικό Υπόβαθρο

Σε αυτό το κεφάλαιο παρουσιάζονται βασικές έννοιες και αλγόριθμοι, οι οποίοι χρησιμοποιούνται εκτενώς στην παρούσα διδακτορική διατριβή. Μετά τον ορισμό του βασικού γραφο-θεωρητικού υποβάθρου, ορίζεται η έννοια της συντομότερης διαδρομής και παρουσιάζεται ο βασικός αλγόριθμος Dijkstra για την εύρεσή τους. Παρουσιάζονται επίσης μέθοδοι αποδοτικής υλοποίησης του αλγορίθμου Dijkstra, καθώς και σημαντικές ευρετικές επεκτάσεις του για την αποδοτική εύρεση συντομότερων διαδρομών από μια αφετηρία προς κάποιον προορισμό.

2.1 Γραφήματα

Ένα *απλό γράφημα* (simple graph) που συμβολίζεται ως $G = (V, E)$ αποτελείται από ένα μη-κενό σύνολο κορυφών (vertices) V και ένα σύνολο E από μη διατεταγμένα ζευγάρια κορυφών του V τα οποία καλούνται πλευρές ή ακμές (edges).

Ένα *διευθυνόμενο ή κατευθυνόμενο γράφημα* (directed graph) $G = (V, E)$ αποτελείται από ένα μη-κενό σύνολο κορυφών V και ένα σύνολο E από διατεταγμένα ζευγάρια κορυφών του V τα οποία καλούνται (κατευθυνόμενες) ακμές ή τόξα. Σε κάθε διευθυνόμενο γράφημα G αντιστοιχεί ένα μη διευθυνόμενο

που προκύπτει από το G αν αγνοηθεί η κατεύθυνση των ακμών.

Αν $e = (u, v)$ είναι μία κατευθυνόμενη ακμή, τότε οι κορυφές u, v καλούνται *άκρα* (end-points) της ακμής. Επίσης η κορυφή u καλείται *ουρά* ή *αρχική κορυφή* (tail or source), ενώ η v καλείται *κεφαλή* ή *τελική κορυφή* (head or target). Στην περίπτωση αυτή συμβολίζουμε με $u = \text{source}(e)$ και με $v = \text{target}(e)$. Ακόμα, η e είναι προσκείμενη (incident) στις u, v , και η (u, v) καλείται εξερχόμενη πλευρά της u και εισερχόμενη της v . Επιπλέον, οι κόμβοι u, v λέγονται γειτονικοί (neighbours). Μία μη-κατευθυνόμενη ακμή (u, v) δεν έχει αρχική ή τελική κορυφή, ενώ ισοδυναμεί με δύο κατευθυνόμενες ακμές (u, v) και (v, u) .

Ο *βαθμός* μιας κορυφής u σε ένα μη-κατευθυνόμενο γράφημα $G = (V, E)$ είναι ο αριθμός των ακμών που προσπίπτουν σε αυτή. Συμβολίζουμε με $\deg(u)$ τον βαθμό της u .

Σε ένα κατευθυνόμενο γράφημα, έχουμε βαθμό εισόδου και εξόδου μιας κορυφής. Ο βαθμός εισόδου μιας κορυφής x , $\text{in-deg}(x)$, είναι ο αριθμός των ακμών στο G που έχουν την x σαν τελική κορυφή. Ο βαθμός εξόδου μιας κορυφής x , $\text{out-deg}(x)$, είναι ο αριθμός των ακμών στο G που έχουν την x σαν αρχική κορυφή.

Ένα γράφημα $H = (V_H, E_H)$ είναι υπογράφημα ενός γραφήματος G όταν $V_H \subseteq V$ και $E_H \subseteq E$. Το H καλείται επαγόμενο (induced) υπογράφημα του G αν $E_H = \{(u, v) \in E \mid u \in V_H \text{ και } v \in V_H\}$

Έστω k ένας μη-αρνητικός ακέραιος και $G = (V, E)$ ένα γράφημα. Μια *διαδρομή* (path) μήκους k από την κορυφή u στην κορυφή v είναι μια ακολουθία κορυφών $u = x_1, \dots, x_k = v$, τέτοια ώστε $(x_i, x_{i+1}) \in E$, $\forall 1 \leq i < k$. Μια διαδρομή είναι κυκλική αν αρχίζει και τελειώνει με την ίδια κορυφή (δηλ. $x_1 = x_k$). Μια διαδρομή είναι απλή αν οι κορυφές $x_1, \dots, x_k$ είναι όλες διακριτές μεταξύ τους.

Ένα μη κατευθυνόμενο γράφημα καλείται *συνεκτικό* (connected) αν υπάρχει διαδρομή μεταξύ κάθε ζευγαριού διακριτών κορυφών στο γράφημα.

Ένα γράφημα ονομάζεται δένδρο όταν είναι συνεκτικό και δεν περιέχει κύκλους. Ένα γεννητικό δένδρο του G είναι ένα υπογράφημα του G που περιέχει όλους τους κόμβους του και είναι δένδρο.

2.2 Συντομότερες Διαδρομές

Σε πολλές εφαρμογές γραφημάτων έχει νόημα η αντιστοίχιση κάθε ακμής με μία τιμή η οποία συνήθως ονομάζεται κόστος ή βάρος. Σε αυτήν την περίπτωση το κατευθυνόμενο γράφημα — που μπορεί να ονομάζεται και δίκτυο (network)— συμβολίζεται συνήθως ως $G = (V, E, wt)$ όπου $wt : E \rightarrow \mathbb{R}$ είναι η συνάρτηση κόστους των ακμών.

Σε αυτή την περίπτωση, ένα κλασικό πρόβλημα είναι το λεγόμενο πρόβλημα *εύρεσης συντομότερης διαδρομής* στο G . Έστω $P = (v_1, \dots, v_k)$ μια διαδρομή του G . Ορίζουμε ως κόστος $wt(P)$ της P το άθροισμα των κοστών των ακμών που την απαρτίζουν, δηλ. $wt(P) = \sum_{i=1}^{k-1} wt(v_i, v_{i+1})$. Δεδομένων δύο κορυφών s και t , η *συντομότερη διαδρομή* (shortest path) $SP(s, t)$ είναι μια διαδρομή από την s στην t με ελάχιστο κόστος μεταξύ όλων των διαδρομών $s-t$. Συμβολίζουμε με $\delta(s, t) = wt(SP(s, t))$ την *απόσταση* της t από την s .

Η σημαντικότερη παραλλαγή του προβλήματος εύρεσης συντομότερων διαδρομών είναι η εύρεση όλων των συντομότερων διαδρομών από μια αρχική κορυφή s προς όλες τις υπόλοιπες κορυφές του G (single-source shortest path). Αν είναι σαφές ποια είναι η αρχική κορυφή s , τότε γράφουμε $\delta(v)$ για την απόσταση $\delta(s, v)$ μιας κορυφής v από την s . Οι συντομότερες διαδρομές από την s προς όλες τις άλλες κορυφές του γραφήματος μπορούν να αποθηκευθούν, με συμπυκνωμένο τρόπο, στο ονομαζόμενο *δένδρο συντομότερων διαδρομών* (shortest path tree) [1]. Ένα δένδρο συντομότερων διαδρομών T είναι ένα δένδρο με ρίζα την s και στο οποίο κάθε δενδρική $s-u$ διαδρομή αντιστοιχεί σε μια συντομότερη $s-u$ διαδρομή στο γράφημα G .

Ένα σημαντικό ζήτημα στα προβλήματα εύρεσης συντομότερων διαδρομών είναι το πρόσημο των κοστών των ακμών. Αρνητικά κόστη μπορεί να οδηγούν σε αρνητικούς κύκλους (δηλ. κύκλων C με $wt(C) < 0$), των οποίων η ύπαρξη αποκλείει την εύρεση συντομότερων διαδρομών αφού στην περίπτωση αυτή οι συντομότερες διαδρομές δεν είναι καλώς ορισμένες [1]. Στα προβλήματα συντομότερων διαδρομών που πραγματεύεται η παρούσα διδακτορική διατριβή όλα τα κόστη ακμών είναι μη αρνητικά. Για το λόγο αυτό, στο υπόλοιπο της διδακτορικής διατριβής θα θεωρείται ότι η συνάρτηση κόστους είναι της μορφής $wt : E \rightarrow \mathbb{R}_0^+$.

2.3 Αλγόριθμος Dijkstra

Ένας από τους πιο γνωστούς αλγόριθμους επίλυσης του προβλήματος εύρεσης συντομότερων διαδρομών από μια αρχική κορυφή s προς όλες τις υπολοιπες σε γραφήματα με μη αρνητικά κόστη ακμών είναι ο αλγόριθμος του Dijkstra [23]. Ως είσοδος δίνεται το κατευθυνόμενο γράφημα $G = (V, E)$ με τη συνάρτηση $wt : E \rightarrow \mathbb{R}_0^+$ κόστους των ακμών, και μια αρχική κορυφή $s \in V$.

Ο αλγόριθμος Dijkstra είναι ένας αλγόριθμος ανάθεσης ετικετών $d[s, u]$ στις κορυφές $u \in V$ του γραφήματος G , οι οποίες θα συγκλίνουν σταδιακά στις αποστάσεις $\delta(s, u)$.

Εφόσον η αρχική κορυφή $s \in V$ είναι σαφώς προσδιορισμένη, για διευκόλυνση στο συμβολισμό, η ετικέτα μιας κορυφής u θα αναφέρεται ως $d[u]$ (αντί $d[s, u]$) και η απόσταση $\delta(s, u)$ της u από την s θα αναφέρεται ως $\delta(u)$.

Πιο συγκεκριμένα, για κάθε κορυφή $u \in V$, ο αλγόριθμος Dijkstra διατηρεί μία ετικέτα $d[u] \geq \delta(u)$. Σε κάθε βήμα, ο αλγόριθμος χρησιμοποιεί δύο σύνολα $S, \bar{S}$, τα οποία και αποτελούν διαμέριση του συνόλου V των κορυφών. Το S περιλαμβάνει τις μόνιμες ή σταθεροποιημένες κορυφές, δηλαδή αυτές στις οποίες έχει ανατεθεί μία μόνιμη ετικέτα, ενώ το $\bar{S}$ περιλαμβάνει τις κορυφές με προσωρινές ετικέτες.

Σε κάθε βήμα, επιλέγεται εκείνη η κορυφή $u \in \bar{S}$ η οποία έχει τη μικρότερη ετικέτα μεταξύ όλων των κορυφών που ανήκουν στο $\bar{S}$, και μονιμοποιείται με την αφαίρεσή της από το $\bar{S}$ και την τοποθέτησή της στο S . Επιπλέον, ανανεώνονται οι ετικέτες των γειτονικών κορυφών, με τη λεγόμενη διαδικασία της «χαλάρωσης» των εξερχόμενων ακμών της· δηλαδή, για κάθε ακμή $(u, v) \in E$, αν $d[v] > d[u] + wt(u, v)$, ο αλγόριθμος θέτει $d[v] = d[u] + wt(u, v)$. Για την ανάκτηση μιας συντομότερης διαδρομής, κάθε κορυφή u διατηρεί έναν δείκτη $pred(u)$ προς την αμέσως προηγούμενη της κορυφή σε μια συντομότερη s - u διαδρομή. Ο ψευδοκώδικας του αλγορίθμου παρουσιάζεται στο Σχήμα 2.1.

Αν $n = |V|$ και $m = |E|$, τότε η πολυπλοκότητα του αλγορίθμου του Dijkstra όπως αυτός παρουσιάζεται στο Σχήμα 2.1, είναι ίση με $O(m + n^2) = O(n^2)$, αφού $m = O(n^2)$. Η χρονική αυτή πολυπλοκότητα προκύπτει από την παρατήρηση ότι κάθε ακμή χαλαρώνεται ακριβώς μία φορά (άρα συνολικός χρόνος $O(m)$ για τη χαλάρωση των ακμών), και από την παρατήρηση ότι θα χρειαστεί

Είσοδος:

$G = (V, E)$ με συνάρτηση κόστους $wt : E \rightarrow \mathbb{R}_0^+$
 $s \in V$ ο κόμβος αφετηρίας

for each $u \in V$ { $d[u] = +\infty; pred(u) = 0;$ }

$d[s] = 0; pred[s] = 0;$

$S = \emptyset; \bar{S} = V;$

while $|S| < |V|$ {

Έστω $u \in \bar{S}$ η κορυφή για την οποία $d[u] = \min_{v \in \bar{S}} d[v]$

$S = S \cup \{u\};$

$\bar{S} = \bar{S} - \{u\};$

for each $(u, v) \in E$ {

if $(d[v] > d[u] + wt(u, v))$ {

$d[v] = d[u] + wt(u, v);$

$pred[v] = u;$

}

}

}

Έξοδος:

$d[\cdot] : \text{πίνακας τέτοιος ώστε } d[u] = \delta(s, u), \forall u \in V$

$pred[\cdot] : \text{δείκτες προγόνων για ανάκτηση συντομότερης διαδρομής}$

Σχήμα 2.1: Ο αλγόριθμος του Dijkstra. S είναι το σύνολο των κορυφών των οποίων οι ετικέτες έχουν σταθεροποιηθεί, ενώ $\bar{S}$ είναι το σύνολο των μη σταθεροποιημένων κορυφών.

να προσδιοριστεί n φορές η κορυφή με τη μικρότερη ετικέτα, από ένα σύνολο το οποίο θα αποτελείται διαδοχικά από $n, n-1, \dots, 1$ στοιχεία, το οποίο συνεπάγεται ότι θα χρειαστεί συνολικός χρόνος $O(n^2)$ για την πραγματοποίηση όλων των επιλογών κορυφών.

2.3.1 Αποδοτική Υλοποίηση Αλγορίθμου Dijkstra

Η ανάλυση της Ενότητας 2.3 καταδεικνύει ότι η χρονική πολυπλοκότητα του Αλγορίθμου Dijkstra καθορίζεται από το βήμα επιλογής κορυφών. Η χρονική πολυπλοκότητα αυτού του βήματος μπορεί να βελτιωθεί με χρήση *ουρών προτεραιότητας*.

Μια *ουρά προτεραιότητας* (priority queue) είναι μια δομή δεδομένων για την αποδοτική αποθήκευση και διαχείριση μιας συλλογής αντικειμένων PQ , κάθε αντικείμενο $x \in PQ$ της οποίας συσχετίζεται με έναν πραγματικό αριθμό ο οποίος καλείται το *κλειδί* ή *προτεραιότητα* του x και συμβολίζεται με $key(x)$.

Μια ουρά προτεραιότητας παρέχει τις εξής λειτουργίες:

- $create(PQ)$: δημιουργεί μια άδεια ουρά προτεραιότητας PQ .
- $findMin(x, PQ)$: βρίσκει και επιστρέφει στη μεταβλητή x το αντικείμενο με το ελάχιστο κλειδί στην PQ .
- $insert(x, PQ)$: εισάγει ένα νέο αντικείμενο x με προκαθορισμένο κλειδί ($key(x)$) στην PQ .
- $decreaseKey(val, x, PQ)$: θέτει $key(x) = val$, όπου $val < key(x)$.
- $deleteMin(x, PQ)$: διαγράφει και επιστρέφει στη μεταβλητή x το αντικείμενο με το ελάχιστο κλειδί στην PQ .

Ο αλγόριθμος Dijkstra μπορεί να υλοποιηθεί με χρήση μιας ουράς προτεραιότητας. Συγκεκριμένα, η ουρά προτεραιότητας υλοποιεί το σύνολο $\bar{S}$ των κορυφών με μη σταθεροποιημένη ετικέτα. Το σύνολο S δεν χρειάζεται να το αποθηκεύσουμε. Το κλειδί μιας κορυφής u θα είναι η ετικέτα της $d[u]$. Ο ψευδοκώδικας του αλγορίθμου Dijkstra με χρήση ουράς προτεραιότητας παρουσιάζεται στο Σχήμα 2.2.

Οποιαδήποτε ουρά προτεραιότητας μπορεί να χρησιμοποιηθεί για την υλοποίηση του αλγορίθμου Dijkstra. Δύο ενδεικτικά παραδείγματα είναι ο b -αδικός σωρός [53, 79] ($b \geq 2$), ο οποίος θα αναφέρεται ως b -σωρός (για $b = 2$ προκύπτει ο κλασικός δυαδικός σωρός [85]), και η ουρά προτεραιότητας Fibonacci [40].

Για μια δεδομένη παράμετρο $b \geq 2$, ο b -σωρός απαιτεί χρόνο $O(\log_b n)$ για τις λειτουργίες $insert$ και $decreaseKey$, χρόνο $O(b \log_b n)$ για τη λειτουργία $deleteMin$, και $O(1)$ χρόνο για τις υπόλοιπες λειτουργίες. Επομένως, η χρονική πολυπλοκότητα του αλγορίθμου είναι $O(m \log_b n + nb \log_b n)$ χρησιμοποιώντας τον b -σωρό. Η πολυπλοκότητα αυτή έχει τη βέλτιστη τιμή της όταν $b = \max\{2, \lceil m/n \rceil\}$, και άρα γίνεται $O(m \log_b n)$. Παρατηρήστε ότι αν

Είσοδος:

$G = (V, E)$ με συνάρτηση κόστους $wt : E \rightarrow \mathbb{R}_0^+$
 $s \in V$ ο κόμβος αφετηρίας

```

for each  $u \in V$  {  $d[u] = +\infty; pred(u) = 0;$  }
 $d[s] = 0; pred[s] = 0;$ 
 $create(PQ); insert(s, PQ);$ 
while  $PQ \neq \emptyset$  {
     $deleteMin(u, PQ);$ 
    for each  $(u, v) \in E$  {
         $val = d(u) + wt(u, w);$ 
        if  $d[v] > val$  then {
            if  $d[v] = \infty$  then
                {  $d[v] = val; insert(v, PQ);$  }
            else
                {  $d[v] = val; decreaseKey(val, v, PQ);$  }
             $pred[v] = u;$ 
        }
    }
}

```

Έξοδος:

$d[\cdot]$: πίνακας τέτοιος ώστε $d[u] = \delta(s, u), \forall u \in V$
 $pred[\cdot]$: δείκτες προγόνων για ανάκτηση συντομότερης διαδρομής

Σχήμα 2.2: Ο αλγόριθμος του Dijkstra με χρήση ουράς προτεραιότητας.

$m = \Omega(n^{1+\varepsilon})$, για κάποια σταθερά $0 < \varepsilon < 1$, η χρονική πολυπλοκότητα του αλγορίθμου είναι βέλτιστη, αφού $O(m \log_b n) = O(m \frac{\log n}{\log b}) = O(m \frac{\log n}{\log n^\varepsilon}) = O(\frac{m}{\varepsilon}) = O(m)$.

Η ουρά προτεραιότητας Fibonacci [40] εκτελεί κάθε λειτουργία σε $O(1)$ επιμερισμένο χρόνο¹ (*amortized time*), εκτός από τη λειτουργία *deleteMin* η οποία απαιτεί επιμερισμένο χρόνο $O(\log n)$. Αξίζει να σημειωθεί ότι έχουν εν τω μεταξύ προκύψει και άλλες αποδοτικές ουρές προτεραιότητας, με χρονικές πολυπλοκότητες ίδιες με εκείνες της ουρά προτεραιότητας Fibonacci και μάλιστα

¹ Η επιμερισμένη πολυπλοκότητα μιας λειτουργίας υποδηλώνει έναν μέσο όρο της πολυπλοκότητας χειρότερης περίπτωσης για την εκτέλεση αυτής της λειτουργίας. Όταν κάποια λειτουργία χρειάζεται επιμερισμένο χρόνο $O(t)$, αυτό σημαίνει ότι μια ακολουθία k τέτοιων λειτουργιών χρειάζεται χρόνο χειρότερης περίπτωσης $O(kt)$.

στη χειρότερη (και όχι στην επιμερισμένη) περίπτωση [13].

Χρησιμοποιώντας μια ουρά προτεραιότητας Fibonacci, είναι εύκολο να διαπιστώσει κανείς ότι η χρονική πολυπλοκότητα του αλγορίθμου Dijkstra γίνεται $O(m + n \log n)$, η οποία αποτελεί τη βέλτιστη (ισχυρά πολυωνυμική) πολυπλοκότητα επίλυσης του προβλήματος συντομότερων διαδρομών από αρχική κορυφή σε κατευθυνόμενο γράφημα με μη αρνητικά κόστη ακμών.

2.4 Ερωτήματα Συντομότερων Διαδρομών και Ευρετικές Μέθοδοι

Ο αλγόριθμος Dijkstra υπολογίζει συντομότερες διαδρομές από μια αρχική κορυφή (*αφετηρία*) s προς όλες τις υπόλοιπες κορυφές του γραφήματος. Πολλές εφαρμογές όμως, απαιτούν τον υπολογισμό ενός *ερωτήματος συντομότερης διαδρομής*, δηλαδή μιας (μόνο) συντομότερης διαδρομής από την αφετηρία s προς μια κορυφή *προορισμό* t .

Το ζητούμενο είναι αν μπορεί αυτός ο υπολογισμός να εκτελεστεί σε χρόνο μικρότερο από εκείνο του αλγορίθμου Dijkstra. Για το σκοπό αυτό έχουν αναπτυχθεί διάφορες ευρετικές τεχνικές επίλυσης ερωτημάτων συντομότερων διαδρομών, οι οποίες είναι πολύ αποδοτικές στην πράξη.

Ο αλγόριθμος Dijkstra εξετάζει τις κορυφές κατά αύξουσα σειρά απόστασης από την s . Διαισθητικά, αναπτύσσει έναν κύκλο με κέντρο την s , το εμβαδό του οποίου αναπαριστάνει τον χώρο αναζήτησης του γραφήματος.

Ένας προφανής τρόπος αποδοτικής επίλυσης ενός ερωτήματος συντομότερης διαδρομής είναι ο πρόωρος τερματισμός του βρόχου `while` μόλις διαγραφεί ο προορισμός t από την ουρά προτεραιότητας. Αν ο t είναι σχετικά «κοντά» (σύμφωνα με τη μετρική που χρησιμοποιείται) στην s , τότε το εμβαδό του κύκλου (χώρος αναζήτησης του γραφήματος) είναι μικρό, και άρα ο χρόνος εκτέλεσης αυτής της ευρετικής τεχνικής θα είναι μικρότερος από το χρόνο εκτέλεσης του αλγορίθμου Dijkstra που διερευνά όλο το γράφημα.

Μια άλλη ευρετική τεχνική είναι εκείνη της *αμφίδρομης αναζήτησης* (*bidirectional search*) [24, 68]: ξεκινά η ταυτόχρονη ανάπτυξη ενός κύκλου Dijkstra από την s και ενός κύκλου από την t (σαρώνοντας εισερχόμενες αντί για ε-

ξερχόμενες ακμές σε κάθε κορυφή), μέχρις ότου βρεθεί μια κοινή ενδιάμεση κορυφή u η οποία αποδεδειγμένα ανήκει τόσο στη συντομότερη $s-u$ διαδρομή, όσο και στην συντομότερη $u-t$ διαδρομή. Η ευρετική τεχνική της αμφίδρομης αναζήτησης, η οποία αναπτύσσει δύο κύκλους, αναμένεται αποδοτικότερη βασισμένη στο διαισθητικό επιχείρημα ότι ένας κύκλος συγκεκριμένης διαμέτρου έχει διπλάσιο εμβαδόν από δύο κύκλους της μισής διαμέτρου.

Στο υπόλοιπο αυτής της ενότητας θα παρουσιαστούν δύο σημαντικές ευρετικές τεχνικές, οι οποίες θα χρησιμοποιηθούν στα πλαίσια της διδακτορικής διατριβής. Οι ευρετικές τεχνικές αυτές ονομάζονται *στοχοκατευθυνόμενες*, επειδή κατευθύνουν ταχύτερα τη διαδικασία αναζήτησης προς τον προορισμό.

2.4.1 Ευρετική Τεχνική A^*

Η βασική ιδέα της ευρετικής τεχνικής A^* [35, 51] είναι να περιορισθεί ο χώρος αναζήτησης του αλγορίθμου Dijkstra, έτσι ώστε να μην αναπτύσσεται ένας κύκλος με κέντρο την αφετηρία s , αλλά μια μικρή ελλειπτική περιοχή με την s ως τη μια εστία έλλειψης και τον προορισμό t σε εκείνο το σημείο της περιμέτρου που είναι εγγύτερα στην άλλη εστία της έλλειψης.

Έστω $\pi_t(u) : V \rightarrow \mathbb{R}$ μια συνάρτηση *δυναμικού* των κορυφών η οποία αναπαριστάνει ένα κάτω φράγμα στην απόσταση μιας κορυφής u προς την t , δηλαδή $\pi_t(u) \leq \delta(u, t)$. Προφανώς, $\pi_t(t) = 0$. Με βάση τα δυναμικά κορυφών, ορίζονται τα *ανηγμένα κόστη* (reduced costs) ακμών: για κάθε ακμή $e = (u, v) \in E$ το ανηγμένο της κόστος $wt'(e)$ ορίζεται ως $wt'(e) = wt(e) - \pi_t(u) + \pi_t(v)$. Είναι σχετικά εύκολο να αποδειχθεί ότι για οποιεσδήποτε δύο κορυφές $x, y \in V$, $\delta'(x, y) = \delta(x, y) - \pi_t(x) + \pi_t(y)$, όπου $\delta'(x, y)$ είναι η απόσταση από την x στην y με βάση τα ανηγμένα κόστη ακμών, και ότι οι συντομότερες διαδρομές, ως προς κόστη $wt'(\cdot)$ και $wt(\cdot)$, ταυτίζονται.

Η ίδια σχέση ισχύει και για τις ετικέτες $d[\cdot]$ που υπολογίζει ο αλγόριθμος Dijkstra (και αποτελούν εκτιμήσεις αποστάσεων μιας κορυφής y από την s), δηλαδή

$$d'[y] = d[y] + \pi_t(y) - \pi_t(s) \quad (2.1)$$

Η ευρετική τεχνική A^* βασίζεται στη χρήση των δυναμικών κορυφών για επιτάχυνση του αλγορίθμου Dijkstra. Κάτι τέτοιο προϋποθέτει ότι τα δυνα-

μικά είναι *έγκυρα*, δηλαδή, για κάθε ακμή $e = (u, v)$, πρέπει να ισχύει ότι $\pi_t(u) - \pi_t(v) \leq wt(e)$. Αυτό συνεπάγεται ότι $wt'(e) \geq 0$, και άρα ο αλγόριθμος Dijkstra μπορεί να εκτελεστεί στο γράφημα με ανηγμένα κόστη ακμών.

Από τα παραπάνω προκύπτει ότι η εκτέλεση του αλγορίθμου Dijkstra με τα ανηγμένα κόστη ακμών υπολογίζει (τις ίδιες με τις αυθεντικές) συντομότερες διαδρομές με ετικέτες $d'[\cdot]$, οι οποίες συνδέονται με τις αυθεντικές ετικέτες $d[\cdot]$ μέσω της σχέσης (2.1).

Ένας εναλλακτικός τρόπος να λάβουμε το ίδιο αποτέλεσμα — αναδιατυπώνοντας τη σχέση (2.1) ως $d'[y] + \pi_t(s) = d[y] + \pi_t(y)$ — προκύπτει από την εκτέλεση του αλγορίθμου Dijkstra στο γράφημα με τα αρχικά κόστη ακμών, αλλά χρησιμοποιώντας ως κλειδί κάθε κορυφής y στην ουρά προτεραιότητας την ποσότητα $d[y] + \pi_t(y)$. Η s εισάγεται στην ουρά προτεραιότητας με κλειδί $\pi_t(s)$. Ο αλγόριθμος αυτός είναι γνωστός ως *ευρετική τεχνική* ή *αναζήτηση A^** .

Η αναζήτηση A^* επιταχύνει την εύρεση μιας συντομότερης s - t διαδρομής, επειδή «κατευθύνει» τον αλγόριθμο Dijkstra και τον χώρο αναζήτησής του κατά μήκος και γύρω από την συντομότερη s - t διαδρομή. Για να γίνει κατανοητό αυτό, θεωρήστε μια συνάρτηση δυναμικού η οποία παράγει τα έγκυρα δυναμικά $\pi_t(u) = \delta(u, t)$, $\forall u \in V$. Τότε, $wt'(e) = wt(e) - \pi_t(u) + \pi_t(v) = wt(e) - \delta(u, t) + \delta(v, t)$, $\forall e = (u, v) \in E$. Αυτό με τη σειρά του σημαίνει ότι για κάθε ακμή $e = (u, v)$ η οποία ανήκει στην συντομότερη s - t διαδρομή, θα ισχύει ότι $wt'(e) = 0$. Άρα ο αλγόριθμος Dijkstra θα ακολουθεί στην περίπτωση αυτή τη διαδρομή με τα μηδενικά ανηγμένα κόστη ακμών, αφού αυτή θα δίνει τις μικρότερες τιμές στις προσωρινές ετικέτες. Από το παράδειγμα αυτό συμπεραίνει κανείς ότι στη γενική περίπτωση ($\pi_t(u) \leq \delta(u, t)$), όσο καλύτερες είναι οι εκτιμήσεις $\pi_t(u)$ κάτω φραγμάτων της $\delta(u, t)$, τόσο γρηγορότερα ο αλγόριθμος Dijkstra θα ανακαλύπτει τη συντομότερη s - t διαδρομή.

2.4.2 Ευρετική Τεχνική ALT

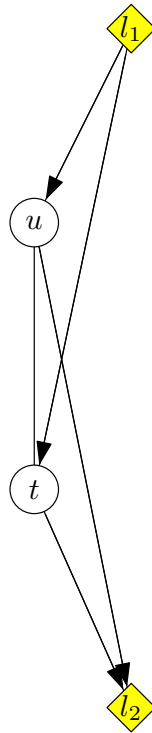
Η ευρετική τεχνική ALT [47] (**A**^{*}, **L**andmarks, and **T**riangle inequality — A^* , ορόσημα, και τριγωνική ανισότητα) αποτελεί μια επέκταση της ευρετικής τεχνικής A^* , η οποία βρίσκει καλές έγκυρες εκτιμήσεις $\pi_t(u)$ κάτω φραγμάτων της $\delta(u, t)$, με χρήση ειδικών κορυφών, που ονομάζονται *ορόσημα*, και χρήση

της τριγωνικής ανισότητας.

Η τριγωνική ανισότητα αναφέρει ότι για οποιεσδήποτε τρεις κορυφές $x, y, \ell \in V$, ισχύει ότι

$$\delta(x, y) \leq \delta(x, \ell) + \delta(\ell, y) \quad (2.2)$$

Έστω τώρα ότι μας δίνεται ένα διακεκριμένο μικρό υποσύνολο κορυφών $L \subset V$, που ονομάζονται *ορόσημα* (landmarks).



Σχήμα 2.3: Ορόσημα και τριγωνική ανισότητα.

Θεωρήστε μια κορυφή u , τον προορισμό t , και δύο ορόσημα $\ell_1, \ell_2 \in L$, όπως στο Σχήμα 2.3. Τότε, από την τριγωνική ανισότητα (2.2), έχουμε ότι:

$$\delta(\ell_1, t) \leq \delta(\ell_1, u) + \delta(u, t)$$

$$\delta(u, \ell_2) \leq \delta(u, t) + \delta(t, \ell_2)$$

Οι παραπάνω ανισότητες μπορούν να αναδιατυπωθούν ως:

$$\delta(\ell_1, t) - \delta(\ell_1, u) \leq \delta(u, t) \quad (2.3)$$

$$\delta(u, \ell_2) - \delta(t, \ell_2) \leq \delta(u, t) \quad (2.4)$$

Τότε, $\forall \ell \in L$, ορίζουμε τις συναρτήσεις δυναμικού

$$\pi_t^{\ell+}(u) = \delta(\ell, t) - \delta(\ell, u) \quad (2.5)$$

$$\pi_t^{\ell-}(u) = \delta(u, \ell) - \delta(t, \ell) \quad (2.6)$$

Από τις ανισότητες (2.3) και (2.4), συνεπάγεται ότι τα δυναμικά $\pi_t^{\ell+}(u)$ και $\pi_t^{\ell-}(u)$ είναι κάτω φράγματα της $\delta(u, t)$ και μπορεί επίσης να αποδειχθεί ότι είναι έγκυρα δυναμικά.

Λήμμα 1 *Οι συναρτήσεις δυναμικού $\pi_t^{\ell+}(u)$ και $\pi_t^{\ell-}(u)$ παράγουν έγκυρα δυναμικά κορυφών.*

Απόδειξη. Δίνεται η απόδειξη για τη συνάρτηση $\pi_t^{\ell+}(u)$. Η απόδειξη για τη συνάρτηση $\pi_t^{\ell-}(u)$ είναι παρόμοια.

Έστω $e = (u, v) \in E$ οποιαδήποτε ακμή. Ζητείται να δειχθεί ότι $wt'(e) = wt(e) - \pi_t^{\ell+}(u) + \pi_t^{\ell+}(v) \geq 0$. Ισχύει ότι:

$$\begin{aligned} wt(e) - \pi_t^{\ell+}(u) + \pi_t^{\ell+}(v) &= \\ wt(e) - (\delta(\ell, t) - \delta(\ell, u)) + (\delta(\ell, t) - \delta(\ell, v)) &= \\ wt(e) + \delta(\ell, u) - \delta(\ell, v) &\geq \\ \delta(u, v) + \delta(\ell, u) - \delta(\ell, v) &\geq 0 \end{aligned}$$

όπου η πρώτη ανισότητα ισχύει επειδή $wt(e) \geq \delta(u, v)$ και η δεύτερη από την τριγωνική ανισότητα (2.2). ■

Επιπλέον, η συνάρτηση $\max$ σε δύο (ή περισσότερα) έγκυρα δυναμικά παράγει επίσης έγκυρα δυναμικά.

Λήμμα 2 *Αν π_t^1 και π_t^2 είναι συναρτήσεις που παράγουν έγκυρα δυναμικά, το ίδιο ισχύει και για τη συνάρτηση $\pi_t = \max\{\pi_t^1, \pi_t^2\}$.*

Απόδειξη. Έστω $e = (u, v) \in E$ οποιαδήποτε ακμή. Εγκυρότητα των π_t^1 και π_t^2 συνεπάγεται ότι:

$$\begin{aligned} wt(e) - \pi_t^1(u) + \pi_t^1(v) &\geq 0 \\ wt(e) - \pi_t^2(u) + \pi_t^2(v) &\geq 0 \end{aligned}$$

Τότε, προφανώς ισχύει ότι:

$$\begin{aligned} wt(e) - \pi_t^1(u) + \max\{\pi_t^1(v), \pi_t^2(v)\} &\geq 0 \\ wt(e) - \pi_t^2(u) + \max\{\pi_t^1(v), \pi_t^2(v)\} &\geq 0 \end{aligned}$$

από τις οποίες έπεται ότι

$$wt(e) - \max\{\pi_t^1(u), \pi_t^2(u)\} + \max\{\pi_t^1(v), \pi_t^2(v)\} \geq 0$$

που αποδεικνύει το ζητούμενο. ■

Το Λήμμα 2 συνεπάγεται το ακόλουθο πόρισμα.

Πόρισμα 1 *Οι συνάρτησεις δυναμικού $\pi_t^\ell(u) = \max\{\pi_t^{\ell+}(u), \pi_t^{\ell-}(u)\}$ και $\pi_t(u) = \max_{\ell \in L} \{\pi_t^\ell(u)\}$ παράγουν έγκυρα δυναμικά.*

Έγκυρες συναρτήσεις δυναμικού μπορούν επίσης να παραχθούν και μέσω άλλων συνδυασμών έγκυρων δυναμικών, για παράδειγμα, μέσω συνάρτησης ελαχίστου (min), μέσου όρου (average), ή γενικά οποιουδήποτε κυρτού συνδυασμού (convex combination) έγκυρων δυναμικών.

Η συνάρτηση (μεγίστου) $\pi_t(u)$ παράγει έγκυρα δυναμικά που δίνουν τα καλύτερα κάτω φράγματα για την απόσταση $\delta(u, t)$.

Όλα τα προαναφερθέντα έγκυρα δυναμικά εξαρτώνται από την επιλογή των οροσήμων. Η εύρεση κατάλληλων οροσήμων, τα οποία να δίνουν όχι μόνο έγκυρα αλλά και τα καλύτερα (ως προς το κάτω φράγμα) δυναμικά, είναι ένα σημαντικό ζήτημα που μελετήθηκε εκτενώς στην εργασία [47]. Τα ορόσημα δεν χρειάζεται να είναι πολλά. Τα ορόσημα πρέπει να είναι όσο το δυνατόν πιο απομακρυσμένα μεταξύ τους και στην «περιφέρεια» του γραφήματος.

Δεδομένου του συνόλου $L \subset V$ των οροσήμων, ο υπολογισμός των δυναμικών παράγεται, σε μια φάση προεπεξεργασίας, με την εύρεση συντομότερων

διαδρομών από κάθε κάθε κορυφή $u \in V$ προς κάθε ορόσημο $\ell \in L$ και αντίστροφα. Τα δυναμικά $\pi_t^{\ell+}(u)$ και $\pi_t^{\ell-}(u)$ μιας κορυφής u υπολογίζονται κατόπιν από τις σχέσεις (2.5) και (2.6), και το τελικό δυναμικό της u υπολογίζεται από τις σχέσεις του Πορίσματος 1.

Κεφάλαιο 3

Γραφοκεντρικά Μοντέλα

Σε αυτό το κεφάλαιο παρουσιάζεται το βασικό μαθηματικό υπόβαθρο αναπαράστασης της πληροφορίας που αφορά πίνακες δρομολογίων μέσω μαζικής μεταφοράς (MMM) και ορίζονται τα θεμελιώδη προβλήματα αναζήτησης και ενημέρωσης πληροφορίας, τα οποία μελετά η παρούσα διδακτορική διατριβή. Παρουσιάζεται ένα βασικό γραφοκεντρικό μοντέλο αναπαράστασης πληροφορίας πινάκων δρομολογίων (ρεαλιστικό χρονοεκτεταμένο μοντέλο), μαζί με τους αλγόριθμους απόκρισης ερωτημάτων αναζήτησης και ενημέρωσης της πληροφορίας. Εν συνεχεία γίνεται μια κριτική επισκόπηση των σημαντικότερων σύγχρονων γραφοκεντρικών δυναμικών μοντέλων αναπαράστασης μεγάλου όγκου δεδομένων. Το κεφάλαιο ολοκληρώνεται με μια αποτίμηση των μοντέλων που παρουσιάζονται.

3.1 Βασικές Έννοιες Αναπαράστασης Πινάκων Δρομολογίων

Ένας *πίνακας δρομολογίων* αποτελείται από σταθμούς, οχήματα (όπως για παράδειγμα, τρένα, λεωφορεία, μετρό, τραμ, ή άλλο MMM), καθώς και τους χρόνους άφιξης και αναχώρησης των οχημάτων στους σταθμούς. Φορμαλιστικά, ένας πίνακας δρομολογίων που αντιστοιχεί σε ένα οποιοδήποτε δημόσιο συγκοινωνιακό δίκτυο MMM περιγράφεται ως εξής.

Ένας πίνακας δρομολογίων $\mathcal{T}$ ορίζεται ως μια τριάδα συνόλων $\mathcal{T} = (\mathcal{Z}, \mathcal{B}, \mathcal{C})$,

όπου $\mathcal{Z}$ είναι το σύνολο των *οχημάτων*, $\mathcal{B}$ το σύνολο των *σταθμών*, και $\mathcal{C}$ το σύνολο των *στοιχειωδών συνδέσεων*.

Μια *στοιχειωδής σύνδεση* είναι μια πλειάδα της μορφής $c = (Z, S_d, S_a, t_d, t_a)$. Μια τέτοια πλειάδα ερμηνεύεται σαν ένα όχημα $Z \in \mathcal{Z}$ που φεύγει από τον σταθμό $S_d \in \mathcal{B}$ την χρονική στιγμή t_d , και η αμέσως επόμενη στάση του οχήματος Z είναι στον σταθμό $S_a \in \mathcal{B}$ την χρονική στιγμή t_a . Εάν x συμβολίζει ένα πεδίο της πλειάδας, τότε με $x(c)$ συμβολίζουμε την τιμή του x στην στοιχειώδη σύνδεση c . Για παράδειγμα, $t_d(c)$ δηλώνει το χρόνο αναχώρησης της c .

Ο χρόνος αναχώρησης $t_d(c)$ και άφιξης $t_a(c)$ μιας στοιχειώδους σύνδεσης c κατά την διάρκεια μιας ημέρας είναι ακέραιοι στο διάστημα $\{0, 1, \dots, 1439\}$ και αναπαριστούν χρόνο σε λεπτά μετά τα μεσάνυχτα. Υποθέτουμε ότι $|\mathcal{C}| \geq \max\{|\mathcal{B}|, |\mathcal{Z}|\}$, αφού δεν μας ενδιαφέρουν οχήματα και σταθμοί οι οποίοι δεν συμμετέχουν σε καμία στοιχειώδη σύνδεση.

Σε δύο δεδομένες χρονικές στιγμές t_1, t_2 , ορίζουμε ως $\Delta(t_1, t_2) = t_2 - t_1 \pmod{1440}$ τη χρονική τους διαφορά, υποθέτοντας ότι η t_2 είναι μεταγενέστερη της t_1 . Το (χρονικό) *μήκος* $\Delta(c)$ μιας στοιχειώδους σύνδεσης c είναι ο χρόνος που μεσολαβεί μεταξύ των στιγμών αναχώρησης και άφιξης της c , υποθέτοντας ότι η c διαρκεί λιγότερο από 24 ώρες· δηλαδή, $\Delta(c) = \Delta(t_d(c), t_a(c))$.

Έστω c_1 μια στοιχειώδη σύνδεση που φτάνει στο σταθμό S και έστω c_2 μια στοιχειώδης σύνδεση που αναχωρεί από τον ίδιο σταθμό S . Αν $Z(c_1) \neq Z(c_2)$, τότε μια μετεπιβίβαση ενός επιβάτη από το $Z(c_1)$ στο $Z(c_2)$ είναι εφικτή αν ο χρόνος μεταξύ της άφιξης και της αναχώρησης στο σταθμό S είναι μεγαλύτερος ή ίσος με τον δεδομένο *ελάχιστο χρόνο μετεπιβίβασης* $transfer(S)$ του σταθμού S . Υποθέτουμε ότι $transfer(S)$ είναι πάντα μικρότερο του 1440, για κάθε $S \in \mathcal{B}$.

Μία *διαδρομή* (ή *ταξίδι*, ή *δρομολόγιο*) σε έναν πίνακα δρομολογίων $\mathcal{T}$ είναι μια ακολουθία στοιχειωδών συνδέσεων $P = (c_1, c_2, \dots, c_k)$, έτσι ώστε, για κάθε $i = 2, 3, \dots, k$, να ισχύει ότι $S_a(c_{i-1}) = S_d(c_i)$ και

$$\Delta(t_a(c_{i-1}), t_d(c_i)) \geq \begin{cases} 0 & \text{αν } Z(c_{i-1}) = Z(c_i) \\ transfer(S_a(c_{i-1})) & \text{αλλιώς} \end{cases}$$

Λέμε ότι η διαδρομή ξεκινά από τον σταθμό $S_d(c_1)$ την χρονική στιγμή $t_d(c_1)$

και φτάνει στον σταθμό $S_a(c_k)$ την χρονική στιγμή $t_a(c_k)$. Το (χρονικό) μήκος $\Delta(P)$ μίας διαδρομής P δίνεται από το άθροισμα των (χρονικών) μηκών των στοιχειωδών συνδέσεων με τους αντίστοιχους χρόνους μετεπιβίβασης· δηλαδή, $\Delta(P) = \sum_{i=1}^k \Delta(c_i) + \Delta(t_a(c_i), t_d(c_i + 1))$.

Ένα *ερώτημα αναζήτησης ή εύρεσης* βέλτιστης (υπό κάποιο/α κριτήριο/α) διαδρομής σε έναν πίνακα δρομολογίων ορίζεται από μια τριάδα (S, T, t_S) , όπου $S \in \mathcal{B}$ είναι ένας σταθμός αναχώρησης, $T \in \mathcal{B}$ είναι ένας σταθμός άφιξης και t_S είναι ο νωρίτερος χρόνος αναχώρησης.

Υπάρχουν δύο φυσικά κριτήρια βελτιστοποίησης που χρησιμοποιούνται για την απάντηση ερωτημάτων. Αναφέρονται στην εύρεση μιας βέλτιστης διαδρομής από τον σταθμό S στον σταθμό T και η οποία ξεκινά μία χρονική στιγμή μεταγενέστερη της t_S , είτε στον ελάχιστο δυνατό χρόνο, είτε με τον ελάχιστο αριθμό μετεπιβιβάσεων οχημάτων.

Τα δύο παραπάνω κριτήρια ορίζουν τα εξής προβλήματα βελτιστοποίησης [72]:

- Το πρόβλημα **Νωρίτερης Άφιξης (NA – Earliest Arrival Problem)** που αφορά στην εύρεση μιας βέλτιστης διαδρομής για έναν επιβάτη που ξεκινά από τον σταθμό S μια χρονική στιγμή μεταγενέστερη της t_S και φτάνει στον σταθμό T το συντομότερο δυνατόν.
- Το πρόβλημα **Ελάχιστου Αριθμού Μετεπιβιβάσεων (EAM – Minimum Number of Transfers Problem)** που αφορά στην εύρεση μιας βέλτιστης διαδρομής για έναν επιβάτη που ξεκινά από τον σταθμό S μια χρονική στιγμή μεταγενέστερη της t_S και φτάνει στον σταθμό T με τον ελάχιστο δυνατό αριθμό μετεπιβιβάσεων οχημάτων.

Ερχόμαστε τώρα στο ζήτημα των καθυστερήσεων στα MMM, οι οποίες συμβαίνουν συχνά και για διάφορους λόγους.

Στο παραπάνω πλαίσιο μοντελοποίησης ενός πίνακα δρομολογίων $\mathcal{T}$ και των στοιχειωδών συνδέσεων, μια *καθυστερήση* που συμβαίνει σε μια στοιχειώδη σύνδεση c μοντελοποιείται ως αύξηση γ λεπτών στον χρόνο άφιξης της c , δηλαδή, $t'_a(c) = t_a(c) + \gamma \pmod{1440}$.

Ο πίνακας δρομολογίων ενημερώνεται σύμφωνα με μια συγκεκριμένη πολιτική που εξαρτάται από την δομή του δικτύου. Ο πίνακας δρομολογίων που προκύπτει ονομάζεται *αναθεωρημένος πίνακας δρομολογίων* (disposition timetable), συμβολίζεται με $\mathcal{T}'$, και διαφέρει από τον $\mathcal{T}$ στους χρόνους άφιξης και αναχώρησης των οχημάτων που εξαρτώνται από το $Z(c)$ στο $\mathcal{T}$. Υπάρχουν πολλές πολιτικές που χρησιμοποιούνται για την ενημέρωση ενός πίνακα δρομολογίων μετά από μια καθυστέρηση οχήματος. Οι βιβλιογραφικές αναφορές [18, 17, 39, 62, 74] περιέχουν πολλά παραδείγματα τέτοιων πολιτικών.

Στην περίπτωση μας, υιοθετούμε την απλούστερη και συνηθέστερη πολιτική καθυστερήσεων που αναφέρει ότι κανένα όχημα δεν περιμένει κάποιο άλλο καθυστερημένο όχημα. Κατά συνέπεια, όταν μία καθυστέρηση συμβαίνει στην σύνδεση c , οι μόνες τιμές χρόνου που επικαιροποιούνται είναι εκείνες που αφορούν τους χρόνους αναχώρησης του $Z(c)$. Επιπρόσθετα, υποθέτουμε ότι η συγκεκριμένη πολιτική δεν λαμβάνει υπόψιν της τυχόν ευελξία στο χρόνο αναχώρησης των οχημάτων (πχ την εσκεμμένη καθυστέρηση αναχώρησης κάποιου οχήματος για την επίτευξη μετεπιβίβασης επιβατών με το καθυστερημένο όχημα) και κατά συνέπεια οι τιμές χρόνου που αφορούν το $Z(c)$ ανανεώνονται με την προσθήκη της τιμής $\gamma \pmod{1440}$.

Το πρόβλημα της *ενημέρωσης* ενός πίνακα δρομολογίων $\mathcal{T}$ αφορά στην ενημέρωση της δομής που αναπαριστάνει τον $\mathcal{T}$, έτσι ώστε τα ερωτήματα εύρεσης βέλτιστων διαδρομών να απαντώνται σωστά. Δηλαδή, η ενημερωμένη δομή πρέπει να αναπαριστάνει τον αναθεωρημένο πίνακα δρομολογίων $\mathcal{T}'$.

3.2 Ρεαλιστικό Χρονοεκτεταμένο Μοντέλο

Σε αυτή την ενότητα, θα ορίσουμε το ρεαλιστικό χρονοεκτεταμένο μοντέλο (Realistic Time-Expanded Model — TE-real) το οποίο παρουσιάστηκε στην εργασία [72], καθώς και τους αλγόριθμους απάντησης ερωτημάτων και ενημέρωσης πίνακα δρομολογίων στην περίπτωση καθυστερήσεων, σε αυτό το μοντέλο.

3.2.1 Μοντέλο

Στο μοντέλο TE-real, ο πίνακας δρομολογίων αναπαρίστανεται ως ένα κατευθυνόμενο γράφημα, το *ρεαλιστικό χρονοεκτεταμένο γράφημα*, που κατασκευάζεται ως εξής: για κάθε στοιχειώδη σύνδεση c δημιουργείται μία κορυφή *αναχώρησης*, μία κορυφή *άφιξης*, και εισάγεται μία ακμή *σύνδεσης* μεταξύ αυτών των δύο κορυφών με κόστος $\Delta(c) = \Delta(t_a(c), t_a(c))$.

Για κάθε γεγονός αναχώρησης, δημιουργείται μία επιπλέον κορυφή, που ονομάζεται κορυφή *μετεπιβίβασης*, η οποία συνδέεται με την αντίστοιχη κορυφή αναχώρησης με μια ακμή *μετεπιβίβασης-αναχώρησης* κόστους 0. Το είδος αυτό των κορυφών απεικονίζουν τις μετεπιβιβάσεις μέσα στον ίδιο τον σταθμό.

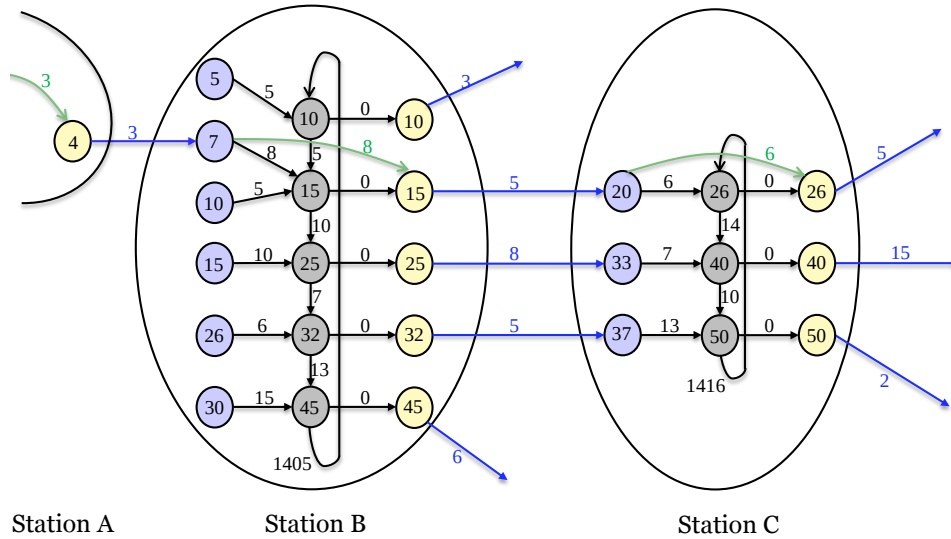
Για μία κορυφή u , έστω $t(u)$ η χρονοσφραγίδα της u στον πίνακα δρομολογίων. Για να διασφαλίσουμε την ικανοποίηση του ελάχιστου χρόνου μετεπιβίβασης σε ένα σταθμό S , εισάγεται μια ακμή *άφιξης-μετεπιβίβασης* από κάθε κορυφή αναχώρησης u στο συντομότερο δυνατό χρονικό στιγμιότυπο που απεικονίζει μια κορυφή μετεπιβίβασης v , ώστε να ισχύει $\Delta(t(u), t(v)) \geq \text{transfer}(S)$.

Για να διασφαλίσουμε την περίπτωση κατά την οποία ένας επιβάτης παραμένει στο ίδιο όχημα όταν διέρχεται από τον σταθμό (χωρίς μετεπιβίβαση), δημιουργούμε μία επιπλέον ακμή (u, v) που ονομάζεται *άφιξης-αναχώρησης* η οποία ενώνει την κορυφή άφιξης u με εκείνη την κορυφή αναχώρησης v που ανήκει στο ίδιο όχημα, και κόστος της ακμής (u, v) ίσο με $\Delta(t(u), t(v))$.

Επιλέον, για να είναι εφικτή η πρόσβαση σε ένα οποιοδήποτε άλλο όχημα, διατάσσουμε τις κορυφές μετεπιβίβασης κατά αύξουσα χρονική σειρά. Με βάση την παραπάνω διάταξη, δύο γειτονικές κορυφές συνδεούνται με ακμή *μετεπιβίβασης-μετεπιβίβασης* από την κορυφή μετεπιβίβασης x που αναπαριστά τη μικρότερη χρονική στιγμή $t(x)$ προς την κορυφή μετεπιβίβασης y που αναπαριστά τη μεγαλύτερη χρονική στιγμή $t(y)$ και κόστος ακμής $\Delta(t(x), t(y))$.

Προκειμένου να μοντελοποιήσουμε μεταμεσονύκτιες μετεπιβιβάσεις μέσα σε ένα σταθμό, εισάγουμε μια *μεταμεσονύχτια ακμή* από την κορυφή μετεπιβίβασης που απεικονίζει το μεγαλύτερο χρονικό στιγμιότυπο t_2 προς την κορυφή μετεπιβίβασης που απεικονίζει το μικρότερο χρονικό στιγμιότυπο t_1 . Το κόστος αυτής της ακμής είναι ίσο με $\Delta(t_2, t_1) + 1440$.

Συνοψίζοντας, στο ρεαλιστικό χρονοεκτεταμένο γράφημα, το κόστος $wt(e)$



Σχήμα 3.1: Το ρεαλιστικό χρονοεκτεταμένο μοντέλο (γράφημα). Οι κορυφές άφιξης, μετεπιβίβασης και αναχώρησης απεικονίζονται με μπλε, γκρι και κίτρινο, αντίστοιχα. Οι ακμές σύνδεσης και άφιξης-μετεπιβίβασης απεικονίζονται με μπλε και πράσινο αντίστοιχα. Οι ακμές μετεπιβίβασης απεικονίζονται με μαύρο χρώμα. Ο ελαχιστος χρόνος μετεπιβίβασης είναι 5 λεπτά.

κάθε ακμής $e = (u, v)$, ορίζεται ως η χρονική διαφορά $\Delta(t(u), t(v))$. Επομένως, για κάθε διαδρομή από μία κορυφή s σε μία άλλη κορυφή a στο γράφημα, το άθροισμα των ακμών των κοστών κατά μήκος της διαδρομής είναι ίσο με την χρονική διαφορά $\Delta(t(s), t(a))$.

Ένα παράδειγμα ρεαλιστικού χρονοεκτεταμένου γραφήματος απεικονίζεται στο Σχήμα 3.1.

Ο χώρος που απαιτεί το ρεαλιστικό χρονοεκτεταμένο γράφημα παρουσιάζεται στο παρακάτω θεώρημα.

Θεώρημα 1 Το ρεαλιστικό χρονοεκτεταμένο γράφημα $G = (V, E)$ που αναπαριστάνει έναν πίνακα δρομολογίων $\mathcal{T} = (\mathcal{Z}, \mathcal{B}, \mathcal{C})$ έχει $n = |V| = 3|\mathcal{C}|$ κορυφές, $4|\mathcal{C}| \leq m = |E| \leq 5|\mathcal{C}|$ ακμές, και απαιτεί χώρο $O(|\mathcal{C}|)$.

Απόδειξη. Από την κατασκευή του γραφήματος, για κάθε στοιχειώδη σύνδεση δημιουργούνται δύο κορυφές, μια κορυφή αναχώρησης σε έναν σταθμό και μια κορυφή άφιξης σε γειτονικό σταθμό. Επίσης, για κάθε κορυφή αναχώρησης δημιουργείται μια κορυφή μετεπιβίβασης στον ίδιο σταθμό. Επομένως,

ο αριθμός των κορυφών είναι $n = 3|\mathcal{C}|$. Όσον αφορά τον αριθμό των ακμών, παρατηρήστε ότι υπάρχουν $|\mathcal{C}|$ ακμές που αντιστοιχούν σε στοιχειώδεις συνδέσεις μεταξύ κορυφών αναχώρησης σε έναν σταθμό και άφιξης σε γειτονικό σταθμό (μπλε ακμές στο Σχήμα 3.1), $|\mathcal{C}|$ ακμές άφιξης-μετεπιβίβασης στον ίδιο σταθμό (μαύρες ακμές στο Σχήμα 3.1), $|\mathcal{C}|$ ακμές μετεπιβίβασης-μετεπιβίβασης στον ίδιο σταθμό (μαύρες ακμές στο Σχήμα 3.1), $|\mathcal{C}|$ ακμές μετεπιβίβασης-αναχώρησης στον ίδιο σταθμό (μαύρες ακμές στο Σχήμα 3.1), και το πολύ $|\mathcal{C}|$ ακμές άφιξης-αναχώρησης στον ίδιο σταθμό (πράσινες ακμές στο Σχήμα 3.1) αφού μπορεί να μην περιλαμβάνουν όλες τις κορυφές άφιξης και αναχώρησης στον ίδιο σταθμό. Η επιπλέον πληροφορία που συσχετίζεται με κορυφές και ακμές απαιτεί $O(1)$ χώρο για κάθε κορυφή ή ακμή. Άρα, ο συνολικός χώρος του ρεαλιστικού χρονοεκτεταμένου γραφήματος είναι $O(|\mathcal{C}|)$. ■

3.2.2 Αλγόριθμος Απάντησης Ερωτημάτων

Δεδομένου ενός ερωτήματος (S, T, t_S) στο ρεαλιστικό χρονοεκτεταμένο γράφημα G , το πρόβλημα NA επιλύεται βρίσκοντας μια συντομότερη διαδρομή από μια αρχική κορυφή $s \in S$ σε μια τελική κορυφή $t \in T$ στο γράφημα G . Ο υπολογισμός αυτός μπορεί να γίνει με κλήση του αλγορίθμου Dijkstra από την κορυφή s και τερματισμό του αλγορίθμου όταν διαγραφεί η κορυφή t από την ουρά προτεραιότητας [6]. Πιο συγκεκριμένα, η s είναι η κορυφή μετεπιβίβασης με τη μικρότερη χρονοσφραγίδα $t(s)$ στον σταθμό S έτσι ώστε $t(s) \geq t_S$. Αν δεν υπάρχει τέτοια κορυφή, τότε η s είναι εκείνη η κορυφή μετεπιβίβασης του σταθμού S για την οποία ο χρόνος $t(s)$ είναι ο ελάχιστος. Η κορυφή t είναι μια κορυφή άφιξης στον σταθμό T με την ελάχιστη απόσταση από την κορυφή s , δηλαδή, η πρώτη κορυφή του T που διαγράφεται από την ουρά προτεραιότητας του αλγορίθμου Dijkstra.

Το πρόβλημα EAM μπορεί επίσης να επιλυθεί εύκολα στο χρονοεκτεταμένο γράφημα. Στην πραγματικότητα, αρκεί να τροποποιηθεί η συνάρτηση κόστους του γραφήματος, θέτοντας κόστος 1 στις ακμές που αναπαριστούν μετεπιβίβαση μέσα σε ένα σταθμό, και θέτοντας κόστος 0 σε οποιαδήποτε άλλη ακμή. Συγκεκριμένα, τα κόστη όλων των εισερχόμενων ακμών των κορυφών μετεπιβίβασης οι οποίες προέρχονται από μια κορυφή άφιξης θέτονται ίσα με 1.

Και στις δύο περιπτώσεις προβλημάτων (NA και EAM), ο αλγόριθμος απάντησης ερωτημάτων έχει χρόνο χειρότερης περίπτωσης $O(|\mathcal{C}| \log |\mathcal{C}|)$ [72]. Στο υπόλοιπο της παρούσας διατριβής, ο παραπάνω αλγόριθμος απάντησης ερωτημάτων θα αναφέρεται ως TE-Q. Η χρονική πολυπλοκότητα του αλγορίθμου καταγράφεται στο παρακάτω θεώρημα.

Θεώρημα 2 [72] *Ο αλγόριθμος TE-Q επιλύει τα προβλήματα NA και EAM σε χρόνο $O(|\mathcal{C}| \log |\mathcal{C}|)$ στο ρεαλιστικό χρονοεκτεταμένο γραφοκεντρικό μοντέλο.*

3.2.3 Αλγόριθμος Ενημέρωσης Καθυστερήσεων

Οι καθυστερήσεις στα συγκοινωνιακά δίκτυα MMM προκύπτουν για διάφορους λόγους· για παράδειγμα, όταν συμβαίνει ένα ατύχημα ή κάποια βλάβη στο δίκτυο ή στο όχημα, κλπ. Η ενημέρωση είναι σημαντική προκειμένου οι επιβάτες να έχουν σωστή πληροφόρηση (πχ στις κινητές συσκευές τους) σχετικά με τον αναθεωρημένο πίνακα δρομολογίων και την ενημερωμένη διαδρομή που πρέπει να ακολουθήσουν.

Σε αυτή την ενότητα θα παρουσιαστεί ο αλγόριθμος ενημέρωσης του ρεαλιστικού χρονοεκτεταμένου μοντέλου (γραφήματος) [26, 43] στην περίπτωση καθυστέρησης οχημάτων σε σχέση με τον δεδομένο πίνακα δρομολογίων. Στο υπόλοιπο της παρούσας διατριβής, ο συγκεκριμένος αλγόριθμος ενημέρωσης θα αναφέρεται ως TE-U.

Η βασική ιδέα του αλγορίθμου είναι η εξής. Όταν σε ένα όχημα συμβαίνει μια καθυστέρηση, δεν αρκεί μόνο η αύξηση του κόστους της ακμής σύνδεσης. Επιπλέον, πρέπει να γίνει ανανέωση των ακμών στο σταθμό που αρχικά έχει συμβεί η καθυστέρηση, αλλά και στους επόμενους σταθμούς που επηρεάζονται από αυτή την καθυστέρηση, με ανανέωση των κοστών συγκεκριμένων ακμών ή/και με επανατοποθέτησή τους (αλλαγή του ενός άκρου τους).

Ορίζουμε μια βασική λειτουργία, τη διαδικασία **επανατοποθέτησης ακμής**: έστω S ένας σταθμός και V_S το σύνολο των κορυφών που αναπαριστούν αφίξεις, μετεπιβιβάσεις και αναχωρήσεις οχημάτων στον S . Επανατοποθέτηση μιας ακμής (x, y) με $x, y \in V_S$, σημαίνει την διαγραφή της (x, y) και την εισαγωγή μιας νέας ακμής (x, z) για κάποιο κατάλληλο $z \in V_S$. Με άλλα λόγια, η ακμή αποκτά μια νέα τελική κορυφή (την z αντί της y).

Ο αλγόριθμος TE-U ενημέρωσης του ρεαλιστικού χρονοεκτεταμένου μοντέλου (γραφήματος) [26, 43] αποτελείται από τρία βήματα.

Πρώτον, ενημερώνεται το κόστος της ακμής που αναπαριστά την άφιξη του οχήματος στον σταθμό. Δεύτερον, ενημερώνεται κατάλληλα το κόστος των ακμών άφιξης-μετεπιβίβασης και μετεπιβίβασης-αναχώρησης στους σταθμούς από τους οποίους διέρχεται το καθυστερημένο όχημα. Τρίτον, γίνεται έλεγχος *εγκυρότητας* των ενημερωμένων ακμών άφιξης-μετεπιβίβασης, δηλαδή αν το κόστος κάθε τέτοιας ακμής είναι μεγαλύτερο από τον χρόνο μετεπιβίβασης του σταθμού.

Πιο λεπτομερειακά, τα βήματα του αλγόριθμου TE-U έχουν ως:

Βήμα 1 (Αύξηση Κόστους Ακμής Σύνδεσης). Εντοπίζεται η σύνδεση που έχει καθυστέρηση, έστω γ λεπτών. Αυξάνεται το κόστος της ακμής άφιξης κατά γ μόνο στον πρώτο σταθμό που έχει συμβεί η καθυστέρηση. Οποιαδήποτε άλλη ακμή άφιξης του ίδιου οχήματος σε επόμενους σταθμούς παραμένει αμετάβλητη.

Βήμα 2 (Ενημέρωση Κόστους Ακμών Σταθμού). Σε όλους τους σταθμούς που διέρχεται το καθυστερημένο όχημα ενημερώνονται τα κόστη των ακμών άφιξης-μετεπιβίβασης και μετεπιβίβασης-αναχώρησης για τις αφίξεις και αναχωρήσεις του καθυστερημένου οχήματος. Πιο συγκεκριμένα, τα κόστη των ακμών άφιξης-μετεπιβίβασης μειώνονται κατά γ , ενώ τα κόστη των ακμών μετεπιβίβασης-αναχώρησης αυξάνονται από 0 σε γ .

Βήμα 3 (Έλεγχος Εγκυρότητας Ακμών Σταθμού). Ελέγχεται κάθε ακμή άφιξης-μετεπιβίβασης που έχει τροποποιηθεί, αν συνεχίζει να είναι έγκυρη· δηλαδή, αν το κόστος της ακμής μετά την αύξηση κατά γ είναι μεγαλύτερο ή ίσο με τον ελάχιστο χρόνο μετεπιβίβασης στον συγκεκριμένο σταθμό. Αν η ακμή είναι έγκυρη, ο έλεγχος συνεχίζεται με την επόμενη τροποποιημένη ακμή άφιξης-μετεπιβίβασης. Αλλιώς, πρέπει να *επανατοποθετηθεί*. Η νέα τελική κορυφή της ακμής θα είναι η αμέσως επόμενη κορυφή μετεπιβίβασης για την οποία η ακμή γίνεται έγκυρη.

Σημειώνεται ότι για τα δύο πρώτα βήματα απαιτείται $O(1)$ και $O(|\mathcal{C}|)$ χρόνος, αντίστοιχα. Για το τρίτο βήμα, απαιτείται $O(|\mathcal{C}| \log |\mathcal{C}|)$ χρόνος, επειδή

χρειάζεται μία δυαδική αναζήτηση στις κορυφές μετεπιβίβασης ώστε να εκτελεστεί ο έλεγχος εγκυρότητας στις ακμές.

Η συζήτηση σε αυτή την ενότητα συνοψίζεται στο παρακάτω θεώρημα.

Θεώρημα 3 *Ο αλγόριθμος $TE-U$ απαιτεί χρόνο $O(|\mathcal{C}| \log |\mathcal{C}|)$ για να ενημερώσει έναν πίνακα δρομολογίων που αναπαριστάνεται από το ρεαλιστικό χρονοεκτεταμένο γραφοκεντρικό μοντέλο μετά από μια καθυστέρηση οχήματος.*

3.3 Άλλα Σύγχρονα Δυναμικά Γραφοκεντρικά Μοντέλα

Το ρεαλιστικό χρονοεκτεταμένο μοντέλο [72] αποτελεί το μοναδικό δυναμικό (αφού υποστηρίζει ενημέρωση) γραφοκεντρικό μοντέλο, το οποίο έχει πλήρως μελετηθεί για αναπαράσταση πληροφορίας πινάκων δρομολογίων, με δυνατότητα ενημέρωσής τους σε περίπτωση καθυστερήσεων [26, 43].

Για λόγους πληρότητας, διερευνήθηκε η σύγχρονη βιβλιογραφία προκειμένου να εξακριβωθούν άλλα σύγχρονα δυναμικά γραφοκεντρικά μοντέλα που πιθανόν να είναι αποδοτικότερα ή πιο εύχρηστα στην αναπαράσταση και ενημέρωση πινάκων δρομολογίων.

Στην ενότητα αυτή παρουσιάζονται τα σχετικότερα τέτοια δυναμικά γραφοκεντρικά μοντέλα. Η απαίτηση υποστήριξης ενημέρωσης σε περίπτωση καθυστερήσεων προσανατολίζει στην χρήση μοντέλων που συντηρούν την μεταβαλλόμενη κατάσταση του υποκείμενου γραφήματος καθώς το γράφημα εξελίσσεται στο χρόνο. Η συλλογή των διαφόρων στιγμιοτύπων (ή καταστάσεων) του γραφήματος σε διαφορετικές χρονικές στιγμές εισάγει την έννοια της *εξελισσόμενης ακολουθίας γραφημάτων*. Σύγχρονα δημοφιλή συστήματα όπως τα Pregel [63], Neo4j [67], Trinity [77], και Giraph [46] εστιάζουν στην επεξεργασία στατικών και πολύ μεγάλων γραφημάτων χωρίς να υποστηρίζουν χρονικές επεκτάσεις των τυπικών ερωτημάτων σε γραφήματα. Ως αποτέλεσμα, αυτά τα συστήματα δεν είναι εγγενώς κατάλληλα για την επεξεργασία και ανάλυση εξελισσόμενων ακολουθιών γραφημάτων.

Η πλειοψηφία της ερευνητικής δραστηριότητας για τη διαχείριση εξελισσόμενων ακολουθιών γραφημάτων στοχεύει στην εκμετάλλευση των κοινών στοι-

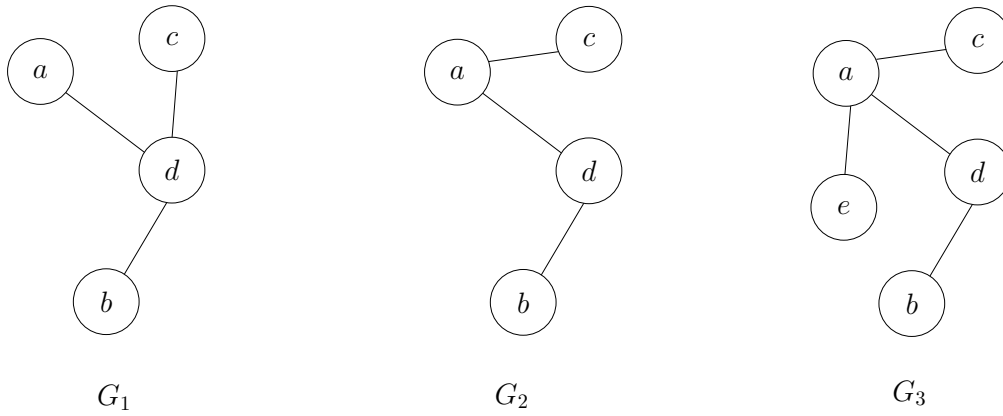
χείων που υπάρχουν στα διαφορετικά χρονικά στιγμιότυπα ενός γραφήματος προκειμένου να βελτιωθεί η χωρική και χρονική αποδοτικότητα. Για παράδειγμα, αν και τα κοινωνικά δίκτυα είναι δυναμικά και αλλάζουν με την πάροδο του χρόνου, η πλειοψηφία των χρηστών και οι μεταξύ τους σχέσεις φιλίας παραμένουν συνήθως αμετάβλητες μεταξύ των πολλαπλών (διαφορετικών) χρονικών στιγμιοτύπων του δικτύου. Για το λόγο αυτό, ένα σύστημα που διαχειρίζεται αποτελεσματικά εξελισσόμενες ακολουθίες γραφημάτων πρέπει να είναι αποδοτικότερο όταν συγκρίνεται με ένα σύστημα επεξεργασίας ενός (μοναδικού) στατικού γραφήματος το οποίο εκτελείται στα διαφορετικά γραφήματα της ακολουθίας.

Η ερευνητική δραστηριότητα στην περιοχή των εξελισσόμενων ακολουθιών γραφημάτων είναι σε αρχικό στάδιο. Για το λόγο αυτό, παρουσιάζονται κεντρικοποιημένα και αποκεντριοποιημένα (παράλληλα και καταναμημένα) μοντέλα.

Μεταξύ των κεντρικοποιημένων μοντέλων ξεχωρίζει το μοντέλο FVF [73], το οποίο ομαδοποιεί τις ακολουθίες γραφημάτων σε συστάδες τις οποίες κατόπιν επεξεργάζεται. Ένα άλλο μοντέλο [55] βασίζεται στη διατήρηση ενός ημερολογίου λειτουργιών (που ορίζεται ως «δέλτα»), οι οποίες εφαρμόζονται στο γράφημα σε διάφορες χρονικές στιγμές και οι οποίες καλούνται για να ανακατασκευάσουν το γράφημα σε μια συγκεκριμένη χρονική στιγμή. Ένα χωρο-αποδοτικό μοντέλο παρουσιάζεται στις εργασίες [11, 15], το οποίο χρησιμοποιεί συμπαγείς και αυτο-δεικτοδοτούμενες δομές δεδομένων για να μειώσει το συνολικό χωρικό κόστος. Κάποια άλλα μοντέλα [2, 52, 87, 76] δημιουργούν ευρετήρια στην ακολουθία γραφημάτων με έναν τρόπο που επιτρέπει τον αποδοτικό υπολογισμό συγκεκριμένων ερωτημάτων.

Όσον αφορά τα αποκεντριοποιημένα μοντέλα, κυριαρχούν δύο προσεγγίσεις: (α) το σύστημα DeltaGraph [54], το οποίο βασίζεται στην έννοια των δέλτα και στοχεύει στην αποδοτική αποθήκευση και ανάκτηση του γραφήματος σε συγκεκριμένες χρονικές στιγμές, και (β) το σύστημα G^* [58, 78, 59], το οποίο είναι μια παράλληλη βάση δεδομένων γραφημάτων η οποία εκμεταλλεύεται τα κοινά στοιχεία μεταξύ των στιγμιοτύπων του γραφήματος σε διαφορετικές χρονικές στιγμές για να αποθηκεύσει χωρο-αποδοτικά την ακολουθία.

Το υπόλοιπο της ενότητας δομείται ως εξής. Αρχικά δίνονται οι βασικοί



Σχήμα 3.2: Μια Εξελισσόμενη Ακολουθία Γραφημάτων.

ορισμοί των εννοιών και προβλημάτων που θα μελετηθούν. Στη συνέχεια αναλύονται τα προαναφερθέντα κεντρικοποιημένα και αποκεντρικοποιημένα γραφοκεντρικά μοντέλα.

3.3.1 Ορισμοί

Σε αυτή την ενότητα θα δωθούν ορισμένοι βασικοί ορισμοί για το γενικό πλαίσιο των προβλημάτων που θα μελετηθούν. Θα ορισθούν τυπικά οι *εξελισσόμενες ακολουθίες γραφημάτων* και στη συνέχεια θα παρουσιασθούν οι διαφορετικοί τύποι ερωτημάτων που μπορούν να εκτελεστούν σε σχέση με τις εξελισσόμενες ακολουθίες γραφημάτων. Χωρίς υστέρηση στη γενικότητα, η παρουσίαση θα επικεντρωθεί στα μη κατευθυνόμενα γράμματα, αφού και τα κατευθυνόμενα γραφήματα ακολουθούν κατά βάση τις ίδιες αρχές.

Ορισμός 1 (Εξελισσόμενη Ακολουθία Γραφημάτων) *Μια εξελισσόμενη ακολουθία γραφημάτων $\mathcal{G}$ είναι μια συλλογή στιγμιότυπων $\mathcal{G} = \langle G_1, G_2, G_3, \dots \rangle$. Ένα στιγμιότυπο γραφήματος $G_i \in \mathcal{G}$, όπου $G_i = (V_i, E_i)$, αντιστοιχεί στο γράφημα G την χρονική στιγμή i και χαρακτηρίζεται από ένα σύνολο κορυφών V_i και ένα σύνολο εισερχόμενων και εξερχόμενων ακμών E_i .*

Ο ρυθμός με τον οποίο λαμβάνονται τα στιγμιότυπα εξαρτάται από το υποκείμενο δίκτυο που αναπαρίσταται από το γράφημα, και σε μεγάλο βαθμό εξαρτάται από την εφαρμογή.

Το Σχήμα 3.2 απεικονίζει μία εξελισσόμενη ακολουθία γραφημάτων, η οποία θα χρησιμεύσει σαν παράδειγμα στο υπόλοιπο του κεφαλαίου. Σε αυτό το παράδειγμα, η εξελισσόμενη ακολουθία γραφημάτων $\mathcal{G}$ αποτελείται από τρία στιγμιότυπα G_1 , G_2 και G_3 με κάθε ένα από αυτά να αντιπροσωπεύει την κατάσταση του γραφήματος G τις χρονικές στιγμές 1, 2 και 3, αντίστοιχα. Για να προκύψει ένα συγκεκριμένο στιγμιότυπο από ένα άλλο στην ακολουθία, πρέπει να εκτελεστεί ένα σύνολο λειτουργιών· για παράδειγμα, προσθέτοντας μία ακμή μεταξύ των κορυφών a και c στο G_1 και αφαιρώντας την ακμή μεταξύ των κορυφών c και d προκύπτει το G_2 . Αξίζει να σημειωθεί ότι ένα σύνολο κορυφών ή ακμών μπορεί να μην αλλάξει καθόλου κατάσταση σε ολόκληρη την ακολουθία (π.χ. η κορυφή b), γεγονός που μπορούμε να εκμεταλλευτούμε ώστε να μειωθεί ο συνολικός χώρος αποθήκευσης της ακολουθίας, ή το χρονικό κόστος απόκρισης ερωτημάτων στην ακολουθία.

Ο σκοπός ενός συστήματος που διαχειρίζεται εξελισσόμενες ακολουθίες γραφημάτων είναι η αποδοτική αποθήκευση ή η δεικτοδότηση (δημιουργία ευρετηρίου) της ακολουθίας, έτσι ώστε να απαντά σε ιστορικά αναλυτικά ερωτήματα. Διακρίνονται δύο εκδοχές του προβλήματος. Στην *στατική* εκδοχή, ολόκληρη η ακολουθία $\mathcal{G}$ είναι γνωστή εκ των προτέρων και οι λειτουργίες ενημέρωσης δεν υποστηρίζονται σε κανένα στιγμιότυπο· δηλαδή, η $\mathcal{G} = \langle G_1, G_2, G_3 \rangle$ αποτελείται μόνο από τα G_1 , G_2 , G_3 και κανένα άλλο νέο στιγμιότυπο δεν δημιουργείται. Στην *δυναμική* εκδοχή, η $\mathcal{G}$ είναι συνεχώς εξελισσόμενη και δεν χαρακτηρίζεται από ένα «τελικό» στιγμιότυπο· δηλαδή, η $\mathcal{G} = \langle G_1, G_2, G_3, \dots \rangle$ μπορεί τελικά να καταλήξει με περισσότερα από τρία στιγμιότυπα.

3.3.2 Τύποι Ερωτημάτων

Τα ερωτήματα που εκτελούνται στις εξελισσόμενες ακολουθίες γραφημάτων μπορούν να χαρακτηριστούν σε σχέση με δύο κύριους τύπους [55] που αφορούν: το *χρονικό πεδίο* (time domain) και την *εμβέλεια γραφήματός* (graph scope) τους.

Όσον αφορά το χρονικό πεδίο, τα ερωτήματα εκτελούνται είτε σε μια συγκεκριμένη χρονική στιγμή, είτε σε ένα χρονικό διάστημα. Στην περίπτωση ενός ερωτήματος χρονικής στιγμής ενδιαφέρει η εξαγωγή ενός χαρακτηριστικού του

γραφήματος τη χρονική στιγμή t , ενώ σε ερώτημα χρονικού διαστήματος ο στόχος είναι η μελέτη της εξέλιξης μιας μετρικής του γραφήματος μέσα σε ένα χρονικό διάστημα $[t, t']$.

Τα ερωτήματα διακρίνονται επίσης από εμβέλεια του γραφήματος στο οποίο λειτουργούν. Πιο συγκεκριμένα, ένα ερώτημα επικεντρώνεται στην αξιολόγηση μιας μετρικής γραφήματος που αφορά είτε ένα μικρό σύνολο κορυφών, είτε ολόκληρο το γράφημα.

Τα περισσότερα ερωτήματα μπορούν να απεικονιστούν σε ένα συνδυασμό των δύο αυτών κατηγοριών. Για παράδειγμα, θεωρήστε το ερώτημα «Πώς έχει εξελιχθεί η συντομότερη απόσταση μεταξύ των κορυφών a και c κατά τις χρονικές στιγμές t_s και t_e », το οποίο μπορεί να οριστεί ως ένα ερώτημα χρονικού διαστήματος που επικεντρώνεται σε ένα σύνολο κορυφών. Ομοίως, το ερώτημα «Ποια είναι η διάμετρος του G τη χρονική στιγμή t_i » είναι ένα ερώτημα μιας χρονικής στιγμής που αφορά ολόκληρο το γράφημα.

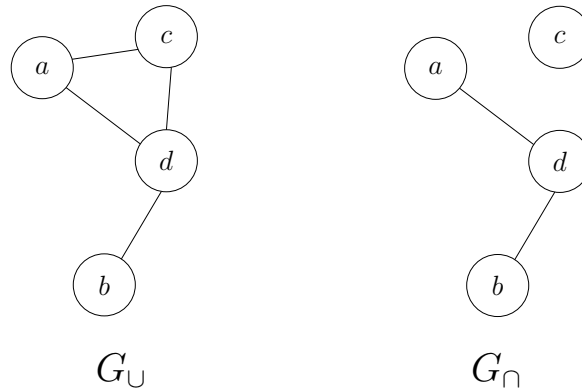
3.3.3 Κεντριοποιημένα Μοντέλα

Στην ενότητα αυτή παρουσιάζονται μοντέλα, μέθοδοι και τεχνικές που αφορούν κεντριοποιημένα περιβάλλοντα. Συγκεκριμένα, παρουσιάζονται τα μοντέλα FVF [73], τα γραφήματα δέλτα [55], και οι συμπαγείς ακολουθίες αναπαράστασης [15]. Η ενότητα ολοκληρώνεται με μεθόδους ευρετηρίων για τις εξελισσόμενες ακολουθίες γραφημάτων, οι οποίες μπορούν να απαντήσουν συγκεκριμένα ιστορικά ερωτήματα.

Το Μοντέλο FVF

Το πρώτο κεντριοποιημένο μοντέλο που εξετάζεται είναι το FVF (Find-Verify-Fix) [73]. Το μοντέλο αυτό αποτελείται από δύο φάσεις: μια πρώτη φάση προεπεξεργασίας και μια δεύτερη φάση επεξεργασίας ερωτημάτων. Επιπρόσθετα, το μοντέλο αυτό υποστηρίζεται από μεθόδους αποθήκευσης εξελισσόμενων ακολουθιών γραφημάτων.

Στην πρώτη φάση της προεπεξεργασίας, τα αρχικά στιγμιότυπα της ακολουθίας ομαδοποιούνται σε μικρότερες συστάδες παρόμοιων στιγμιότυπων. Αυτό γίνεται με τον καθορισμό μιας μετρικής ομοιότητας γραφημάτων και με την



Σχήμα 3.3: Τα γραφήματα G_U και $G_\cap$ για την εξελισσόμενη ακολουθία γραφημάτων του Σχήματος 3.2.

σταδιακή πρόσθεση στιγμιότυπων στην συστάδα (ξεκινώντας από το πρώτο στιγμιότυπο της ακολουθίας) μέχρις ότου να ξεπεραστεί ένα συγκεκριμένο κατώφλι ομοιότητας γραφημάτων. Σε αυτό το σημείο, μια καινούργια άδεια συστάδα δημιουργείται και η παραπάνω διαδικασία επαναλαμβάνεται μέχρις ότου να εξεταστούν όλα τα στιγμιότυπα γραφημάτων. Για κάθε συστάδα, δύο αντιπροσωπευτικά γραφήματα $G_\cap$ και G_U εξάγονται, τα οποία είναι το μεγαλύτερο κοινό υπογράφημα και το μικρότερο κοινό υπογράφημα όλων των στιγμιότυπων της συστάδας, αντίστοιχα. Για παράδειγμα, εάν υποθέσουμε ότι τα γραφήματα G_1 και G_2 του Σχήματος 3.2 ομαδοποιούνται στην ίδια συστάδα, τότε τα αντίστοιχα γραφήματα $G_\cap$ και G_U αντιστοιχούν στα γραφήματα του Σχήματος 3.3.

Στην δεύτερη φάση επεξεργασίας ερωτημάτων, οι συστάδες και τα συσχετιζόμενα γραφήματά τους χρησιμοποιούνται για να απαντήσουν ερωτήματα συντομότερων διαδρομών και κεντρικής εγγύτητας (closeness centrality). Αρχικά, αξιολογείται η λύση σε ένα ερώτημα για τα συσχετιζόμενα γραφήματα μιας συστάδας (βήμα “Find”) στη βάση ότι η λύση θα εφαρμοστεί εύκολα σε έναν αριθμό στιγμιότυπων στην συστάδα. Στο βήμα επιβεβαίωσης (Verify), η αξιολογηθείσα λύση ελέγχεται για κάθε στιγμιότυπο της συστάδας σε συνδυασμό με ένα σύνολο διαισθητικών λημμάτων. Για κάθε στιγμιότυπο, για το οποίο η αξιολογηθείσα λύση δεν εφαρμόζεται, το μοντέλο προσπαθεί να διορθώσει (βήμα Fix) τη λύση, έτσι ώστε να γίνει εφαρμόσιμη στο προαναφερθέν στιγμιότυπο.

Στην εργασία [73], παρουσιάζονται τρεις μέθοδοι αποθήκευσης οι οποίες

μπορούν να χρησιμοποιηθούν μαζί με το μοντέλο FVF. Οι μέθοδοι κάνουν χρήση των ομοιοτήτων που παρουσιάζονται μεταξύ διαδοχικών στιγμιοτύπων και μεταξύ συσχετιζόμενων γραφημάτων διαδοχικών συστάδων για τη μείωση του συνολικού κόστους χώρου της εξελισσόμενης ακολουθίας γραφημάτων. Το μοντέλο FVF αξιολογείται με εκτενή πειράματα σε πραγματικά και συνθετικά σύνολα δεδομένων [73].

Γραφήματα Δέλτα

Το μοντέλο γραφημάτων «δέλτα» [55] αποτελεί ένα μοντέλο υποστήριξης παρελθοντικών (ιστορικών) ερωτημάτων σε ακολουθίες εξελισσόμενων γραφημάτων. Καταρχήν, δηλώνονται οι λειτουργίες οι οποίες υποστηρίζονται σε κάθε στιγμιότυπο, συγκεκριμένα, οι λειτουργίες $addNode(u_i)$, $remNode(u_i)$, $addEdge(u_i, u_j)$, και $remEdge(u_i, u_j)$, οι οποίες αντιστοιχούν στην πρόσθεση ή στην αφαίρεση μιας κορυφής u_i , και στην πρόσθεση ή στην αφαίρεση μιας ακμής μεταξύ δύο κορυφών u_i και u_j , αντίστοιχα. Το *γράφημα δέλτα* ορίζεται ως το σύνολο τέτοιων λειτουργιών, οι οποίες όταν εφαρμόζονται σε ένα συγκεκριμένο στιγμιότυπο παράγουν ένα άλλο στιγμιότυπο της ακολουθίας. Για παράδειγμα, στο Σχήμα 3.2 το G_3 μπορεί να δημιουργηθεί από την εφαρμογή των λειτουργιών $\{addNode(e), addEdge(a, e)\}$ στο G_2 .

Το *πλήρες* (γράφημα) *δέλτα* ορίζεται ως ένα σύνολο λειτουργιών που όταν εφαρμόζονται στο πρώτο στιγμιότυπο της ακολουθίας είναι σε θέση να παράγουν οποιοδήποτε άλλο στιγμιότυπο της ακολουθίας¹. Επιπρόσθετα, το *ανεστραμμένο δέλτα* ορίζεται ως ένα σύνολο λειτουργιών οι οποίες όταν εφαρμοστούν σε ένα στιγμιότυπο G_t παράγεται ένα άλλο στιγμιότυπο $G_{t'}$ για το οποίο $t' < t$, δηλαδή, το $G_{t'}$ εμφανίζεται «νωρίτερα» στην ακολουθία από το G_t .

Έχοντας καθορίσει τους διαφορετικούς τύπους δέλτα, παρουσιάζονται στη συνέχεια τεχνικές και πολιτικές υλοποίησης στιγμιότυπων [55]. Ειδικότερα, ενώ οποιοδήποτε από τα στιγμιότυπα μιας ακολουθίας μπορεί να ανακατασκευαστεί εάν διατηρηθεί ένα πλήρες και αναστρέψιμο δέλτα καθώς και ένα άλλο στιγμιότυπο της ακολουθίας, μπορεί να είναι αποδοτικότερο για τη μέθο-

¹Ορισμένα στιγμιότυπα απαιτούν την εφαρμογή μόνο ενός υποσυνόλου των λειτουργιών σε ένα πλήρες δέλτα.

δο να διατηρηθούν και άλλα παρεμβαλλόμενα στιγμιότυπα για να επιταχυνθεί η υλοποίηση ενός στιγμιότυπου.

Όσον αφορά την αποδοτική επεξεργασία ερωτημάτων, παρουσιάζονται στην εργασία [55] τρεις μέθοδοι. Ίσως η πιο καθολική από τις προτεινόμενες μεθόδους είναι μια διαδικασία ερωτημάτων δύο φάσεων, η οποία πρώτα υλοποιεί ένα συγκεκριμένο στιγμιότυπο σύμφωνα με τις τεχνικές που παρουσιάστηκαν και στη συνέχεια εκτελεί το ερώτημα στο υλοποιημένο στιγμιότυπο. Τέλος, διευρυνώνται πιθανές βελτιστοποιήσεις, προσεγγίσεις δεικτοδότησης των δέλτα, και παρουσιάζονται κάποια προκαταρκτικά αποτελέσματα των προτεινόμενων λύσεων [55].

Αναπαράσταση Συμπαγών Ακολουθιών

Τα μοντέλα που έχουν έως τώρα παρουσιαστεί επικεντρώθηκαν στη μείωση του συνολικού χρόνου των ερωτημάτων στις εξελισσόμενες ακολουθίες γραφημάτων. Στην ενότητα αυτή παρουσιάζεται το μοντέλο της εργασίας [15], στην οποία μελετάται το πρόβλημα της μείωσης του χωρικού κόστους κατά τη διαχείριση των εξελισσόμενων ακολουθιών γραφημάτων.

Οι μέθοδοι που προτείνονται βασίζονται κατά κύριο λόγο σε συμπαγείς και αυτο-δεικτοδοτούμενες δομές δεδομένων, οι οποίες όταν συνδυάζονται με συγκεκριμένες τεχνικές συμπίεσης (όπως την τεχνική ETDC [12] και την τεχνική PForDelta [88, 89]) επιτυγχάνουν μια συνολικά υψηλή χωρο-αποδοτικότητα με μια καλή εξισορρόπηση του συνολικού χρονικού κόστους απάντησης ερωτημάτων.

Το μοντέλο της εργασίας [15] στηρίζεται στην έννοια των *επαφών*, όπως παρουσιάστηκαν και χρησιμοποιήθηκαν στην εργασία [69] για να ορίσουν τα χρονομεταβαλλόμενα γραφήματα² (temporal graphs). Μία *επαφή* ορίζεται ως μία 4-άδα (u, v, t_s, t_e) που υποδηλώνει την ύπαρξη μιας ακμής μεταξύ των κορυφών u και v κατά τη διάρκεια της χρονικής περιόδου $[t_s, t_e]$. Η συλλογή όλων των επαφών είναι ισοδύναμη με το (συνολικό) χρονομεταβαλλόμενο γράφημα, ενώ ένα συγκεκριμένο στιγμιότυπο G_t αντιστοιχεί σε ένα σύνολο επαφών

²Στο υπόλοιπο αυτού του κεφαλαίου, οι όροι «εξελισσόμενη ακολουθία γραφημάτων» και «χρονομεταβαλλόμενο γράφημα» θα χρησιμοποιούνται εναλλακτικά.

(u, v, t_s, t_e) έτσι ώστε $t \in [t_s, t_e]$.

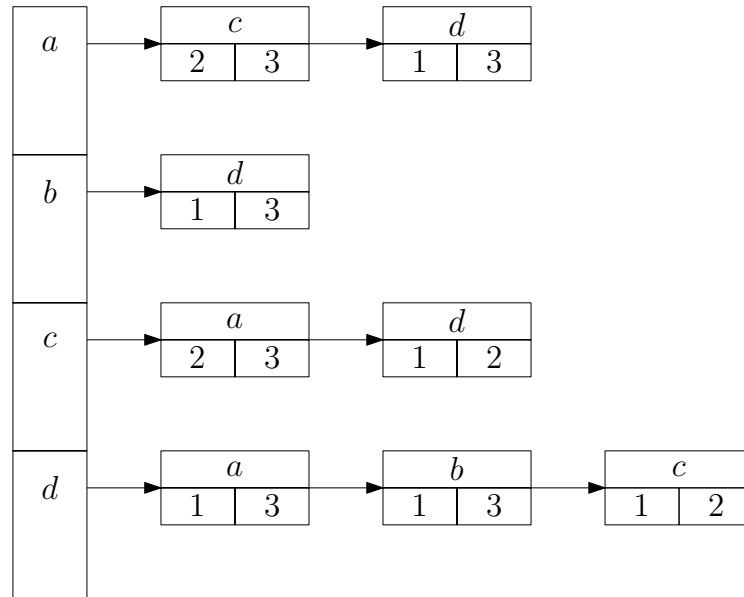
Στη συνέχεια, παρουσιάζονται οι λειτουργίες που μπορούν να εκτελεστούν στα χρονομεταβαλλόμενα γραφήματα. Αυτές περιλαμβάνουν:

- Ερωτήματα γειτνίασης: αναφορά όλων των γειτόνων μιας κορυφής u .
- Αντίστροφα ερωτήματα γειτνίασης: αναφορά όλων των κορυφών που έχουν μια κορυφή u ως γειτονική.
- Ερωτήματα ενεργών ακμών: υπάρχει ακμή μεταξύ δύο κορυφών u και v τη χρονική στιγμή t ;
- Ανάκτηση στιγμιοτύπου γραφήματος τη χρονική στιγμή t .
- Ερωτήματα αλλαγής κατάστασης ακμών: αναφορά όλων των ακμών που άλλαξαν κατάσταση τη χρονική στιγμή t , δηλαδή, όλες τις επαφές με $t_s = t$ ή $t_e = t$.

Μετά από μια σύντομη επισκόπηση των τεχνικών συμπίεσης και των συμπαγών δομών δεδομένων που χρησιμοποιούνται, η εργασία [15] επικεντρώνεται στις τέσσερις χρονομεταβαλλόμενες αναπαραστάσεις γραφημάτων που προτείνονται μαζί με τις υλοποιήσεις τους, οι οποίες εκμεταλλεύονται τις τεχνικές συμπίεσης.

Η πρώτη αναπαράσταση, που ονομάζεται EdgeLog, είναι ένα ευρετήριο που διατηρεί για κάθε κορυφή v στο χρονομεταβαλλόμενο γράφημα μια λίστα γειτονικών κορυφών της v . Κάθε γείτονας της v είναι επίσης εφοδιασμένος με μία λίστα που περιέχει όλα τα χρονικά διαστήματα για τα οποία υπάρχει η συγκεκριμένη ακμή στην ακολουθία. Η δομή EdgeLog για μία ακολουθία που αποτελείται από τα γραφήματα G_1 και G_2 του παραδείγματος του Σχήματος 3.2, απεικονίζεται στο Σχήμα 3.4.

Η δεύτερη αναπαράσταση ονομάζεται EneLog και ακολουθεί την ίδια προσέγγιση με την πρώτη. Πιο συγκεκριμένα, η δομή EneLog αποτελείται από μία λίστα όλων των κορυφών που εμφανίζονται στο χρονομεταβαλλόμενο γράφημα. Για κάθε κορυφή v , υπάρχει μία λίστα όλων των «γεγονότων» που συσχετίζονται με την κορυφή v · δηλαδή, αλλαγές κατάστασης της ακμής μαζί με την τελική της κορυφή.



Σχήμα 3.4: Οι δομές EdgeLog και EveLog της εξελισσόμενης ακολουθίας γραφημάτων του Σχήματος 3.2.

Η τρίτη αναπαράσταση καλείται *συμπαγής ακολουθία γειτονικότητας* (Compact Adjacency Sequence – CAS) και στηρίζεται στην χρήση του δένδρου Wavelet.

Η τέταρτη αναπαράσταση CET βασίζεται στο δένδρο Interleaved Wavelet, το οποίο είναι μία δομή δεδομένων που παρουσιάστηκε στην ίδια εργασία [15] σαν μια επιπρόσθετη δομή αποδοτικού χειρισμού χρονομεταβαλλόμενων γραφημάτων.

Μια εκτεταμένη πειραματική αξιολόγηση σε πραγματικά αλλά και συνθετικά σύνολα δεδομένων οδηγεί στο ενδιαφέρον συμπέρασμα ότι δεν υπάρχει μία και μοναδική δομή δεδομένων για όλα τα ερωτήματα που εκτελούνται σε χρονομεταβαλλόμενα γραφήματα. Ως τελευταία παρατήρηση, πρέπει να σημειωθεί ότι αν και οι μέθοδοι της εργασίας [15] επικεντρώνονται στην στατική εκδοχή του προβλήματος, αναφέρονται ωστόσο τροποποιήσεις και επεκτάσεις που πρέπει να γίνουν ώστε να εφαρμοστούν οι λύσεις και στην δυναμική εκδοχή του προβλήματος.

Δημιουργία Ευρετηρίων για Συγκεκριμένα Ερωτήματα

Τα μοντέλα και οι μέθοδοι που έχουν μέχρι τώρα παρουσιαστεί στην Ενότητα 3.3 επικεντρώνονται κυρίως στην αποτελεσματική αποθήκευση, συντήρηση και ανάκτηση των στιγμιότυπων μιας εξελισσόμενης ακολουθίας γραφημάτων.

Έχουν προταθεί, όμως, μέθοδοι στη βιβλιογραφία που στοχεύουν σε μια διαφορετική κατεύθυνση, αυτή της δεικτοδότησης της εξελισσόμενης ακολουθίας γραφημάτων κατά τρόπο που να επιτρέπει την αποτελεσματική αξιολόγηση συγκεκριμένων ερωτημάτων. Στην ενότητα αυτή παρουσιάζονται ορισμένα αξιοσημείωτα παραδείγματα.

Στην εργασία [2] περιγράφεται η δημιουργία δυναμικών ευρετηρίων που επιτρέπει την απάντηση ερωτημάτων απόστασης είτε στο τελευταίο στιγμιότυπο (τρέχον), είτε σε οποιοδήποτε «παλαιότερο» στιγμιότυπο της ακολουθίας. Επιπλέον, τα ευρετήρια αυτά υποστηρίζουν το ιστορικό ερώτημα απόστασης μεταβλητού σημείου (*historical distance change-point query*), το οποίο αναφέρει όλες τις χρονικές στιγμές στην ακολουθία για τις οποίες αλλάζει η απόσταση μεταξύ δύο κορυφών u και v . Αξίζει να σημειωθεί ότι η εργασία [2] αφορά γραφήματα που υποστηρίζουν μόνο την προσθήκη κορυφών και ακμών.

Μια άλλη μέθοδος που επικεντρώνεται στην απάντηση ερωτημάτων εύρεσης συντομότερων διαδρομών προτάθηκε στην εργασία [52]. Στην εργασία αυτή χρησιμοποιείται η δομή των *χρονομεταβαλλόμενα εξελισσόμενων γραφημάτων* (*temporally evolving graph*) για την αποθήκευση όλων των ενημερώσεων που συμβαίνουν στην ακολουθία. Για τον υπολογισμό των συντομότερων διαδρομών εφαρμόζονται παραλλαγές του αλγορίθμου Dijkstra [23]. Επιπρόσθετα, η εύρεση συντομότερων διαδρομών επιταχύνεται με χρήση ευρετηρίων που δημιουργούνται σε μια φάση προ-επεξεργασίας. Πιο συγκεκριμένα, ως ευρετήριο χρησιμοποιούνται οι ιεραρχίες συρρίκνωσης (*contraction hierarchies* [41]).

Στην εργασία [87] προτείνεται ένας αλγόριθμος που ανακαλύπτει τις συχνότερα μεταβαλλόμενες συνιστώσες σε μια εξελισσόμενη ακολουθία γραφημάτων. Ορίζονται μετρικές για την καταμέτρηση του μεγέθους μεταβολής (αλλαγής) μεταξύ κορυφών καθώς και για το γενικό πρόβλημα της (μέτρησης της) εξαγωγής της συχνότερα μεταβαλλόμενης συνιστώσας, και με βάση αυτές παρουσιάζονται μέθοδοι επίλυσης των προβλημάτων.

Τέλος, στην εργασία [76] μελετάται το πρόβλημα της απάντησης παρελθοντικών (ιστορικών) ερωτήματων προσπελασιμότητας (δηλαδή, αν υπάρχει διαδρομή μεταξύ δύο κορυφών). Η προτεινόμενη δομή ευρετηρίου ονομάζεται TimeReach και είναι κατασκευασμένη έτσι ώστε να εκμεταλλεύεται τις ισχυρά συνεκτικές συνιστώσες του γραφήματος.

3.3.4 Αποκεντριοποιημένα Μοντέλα

Σε αυτή την ενότητα παρουσιάζονται συστήματα, μοντέλα, μέθοδοι και λύσεις που αναπτύχθηκαν για αποκεντριοποιημένα (παράλληλα και κατανεμημένα) περιβάλλοντα. Τα δύο συστήματα/μοντέλα που θα αναλυθούν είναι το DeltaGraph [54] και η παράλληλη βάση δεδομένων γραφημάτων G^* [58, 59, 78].

Το Σύστημα DeltaGraph

Στην εργασία [54] παρουσιάζεται ο σχεδιασμός και η υλοποίηση ενός κατανεμημένου συστήματος που ονομάζεται DeltaGraph, το οποίο στοχεύει στην αποτελεσματική αποθήκευση και ανάκτηση στιγμιότυπων μιας εξελισσόμενης ακολουθίας γραφημάτων. Το σύστημα DeltaGraph υποστηρίζει ερωτήματα μοναδικής χρονικής στιγμής ή σημείου (single-point queries), ερωτήματα στιγμιότυπων για ένα χρονικό διάστημα, και ερωτήματα πολλαπλών χρονικών στιγμών ή σημείων (multi-point queries). Επιπλέον, μαζί με την υποστηριζόμενη δομή γραφημάτων, ένα ερώτημα μπορεί να επιστρέφει κατηγορήματα συσχετιζόμενα με κορυφές και ακμές (π.χ. όνομα, βάρος κ.λ.π.).

Το σύστημα αποτελείται από δύο βασικές συνιστώσες: τη δομή ευρετηρίου DeltaGraph και τη δομή δεδομένων κύριας μνήμης GraphPool.

Η δομή ευρετηρίου DeltaGraph περιγράφεται ως μία έρριξη ιεραρχική δομή γραφήματος (rooted hierarchical graph structure) που παρομοιάζει με δένδρο του οποίου τα γειτονικά φύλλα είναι συνδεδεμένα μεταξύ τους με αμφίδρομο τρόπο. Τα φύλλα της δομής αντιστοιχούν σε στιγμιότυπα της ακολουθίας, ενώ οι εσωτερικοί κόμβοι αντιστοιχούν σε γραφήματα που μπορούν να δημιουργηθούν εφαρμόζοντας μια διαφορετική λειτουργία (π.χ. τομή) στα παιδιά τους. Οι ακμές μεταξύ των κόμβων αποθηκεύουν σύνολα από δέλτα που

χρησιμοποιούνται για τη δημιουργία μιας κορυφής-παιδιού από τον γονέα του, και διαμερίζονται οριζόντια μεταξύ των διαθέσιμων υπολογιστικών νημάτων (workers). Θα πρέπει να σημειωθεί σε αυτό το σημείο ότι τα μόνα δεδομένα που είναι αποθηκευμένα είναι τα σύνολα των γραφημάτων δέλτα και όχι τα ίδια τα γραφήματα, αν και η εργασία [54] προτείνει την υλοποίηση συγκεκριμένων στιγμιότυπων στο DeltaGraph, έτσι ώστε να επιταχυνθεί ο χρόνος απόκρισης των ερωτημάτων.

Για την απάντηση ενός ερωτήματος μοναδικού σημείου που αφορά τη χρονική στιγμή t , το σύστημα εντοπίζει μέσω δυαδικής αναζήτησης μεταξύ των φύλλων τα δύο γειτονικά φύλλα που «περιβάλλουν» τη χρονική στιγμή t . Στη συνέχεια, βρίσκει την συντομότερη διαδρομή από την ρίζα προς ένα από τα δύο φύλλα, όπου το κόστος μιας ακμής είναι ίσο με το μέγεθος των συσχετιζόμενων δέλτα. Για την απάντηση ερωτημάτων πολλαπλών σημείων, το σύστημα ακολουθεί την ίδια διαδικασία με τη διαφορά ότι αντί να βρει μια συντομότερη διαδρομή, το σύστημα πρέπει να βρει ένα δέντρο Steiner μικρότερου κόστους μεταξύ της ρίζας και των πολλαπλών χρονικών στιγμών.

Η άλλη συνιστώσα του συστήματος είναι η δομή δεδομένων GraphPool, η οποία διατηρεί στην κύρια μνήμη έναν συνδυασμό υλοποιημένων στιγμιότυπων. Πιο συγκεκριμένα, η δομή GraphPool διατηρεί το τρέχον γράφημα, τα ιστορικά στιγμιότυπα, και τα υλοποιημένα γραφήματα σε ένα μοναδικό συνδυασμένο γράφημα. Για τον προσδιορισμό των γραφημάτων που περιέχουν μια συγκεκριμένη συνιστώσα ή ένα συσχετισμένο κατηγορημα, το σύστημα χρησιμοποιεί έναν πίνακα αντιστοίχισης. Τέλος, η δομή GraphPool είναι υπεύθυνη για την ενημέρωση του ευρετηρίου του τρέχοντος γραφήματος και την εκκαθάριση των ιστορικών στιγμιότυπων που δεν χρειάζονται άλλο.

Η Βάση Δεδομένων Γραφημάτων G^*

Το τελευταίο σύστημα που εξετάζεται είναι η βάση δεδομένων γραφημάτων G^* [58, 78], η οποία επικεντρώνεται στην αξιοποίηση των κοινών στοιχείων που υπάρχουν μεταξύ των στιγμιότυπων της ακολουθίας, έτσι ώστε να αποθηκεύονται με αποδοτικό τρόπο.

Το σύστημα G^* αναθέτει σε κάθε εξυπηρετητή ένα σύνολο κορυφών μαζί με

τις εξερχόμενες ακμές τους. Αυτό επιτυγχάνει σημαντική τοπικότητα δεδομένων, αφού η ανάκτηση όλων των ακμών μιας κορυφής μπορεί να επιτευχθεί χωρίς να απαιτείται επικοινωνία με οποιονδήποτε από τους άλλους εξυπηρεητές. Επειδή τα στιγμιότυπα σε μια ακολουθία παρουσιάζουν ομοιότητες μεταξύ τους, το σύστημα G^* αποφεύγει την αποθήκευση περιττών πληροφοριών, αποθηκεύοντας κάθε εκδοχή μιας κορυφής μόνο μια φορά. Με αυτό τον τρόπο τα δεδομένα που δεν τροποποιούνται μεταξύ διαφορετικών στιγμιότυπων δεν χρειάζεται να αποθηκευθούν ξανά.

Κάθε εξυπηρετητής διατηρεί ένα ευρετήριο, που καλείται *συμπαγές ευρετήριο γραφήματος* (compact graph index – CGI), το οποίο αποθηκεύει ένα μοναδικό ζευγάρι ($vertexID, disk_location$) για κάθε εκδοχή μιας κορυφής, η οποία υπάρχει σε ένα συνδυασμό στιγμιότυπων της ακολουθίας. Για παράδειγμα, το ευρετήριο CGI ενός εξυπηρετητή που διατηρεί την κορυφή c του Σχήματος 3.2 θα μπορούσε να περιέχει δύο ζευγάρια που σχετίζονται με την c : ένα ζευγάρι για την εκδοχή c_1 στο $\{G_1\}$ και ένα άλλο για την εκδοχή c_2 στα $\{G_2, G_3\}$. Πρέπει να σημειωθεί ότι το ευρετήριο CGI έχει μικρή επιβάρυνση χώρου και μπορεί πλήρως ή κατά το μεγαλύτερο μέρος του να αποθηκευθεί στην κύρια μνήμη. Για λόγους αποδοτικής διαχείρισης, όταν το περιεχόμενο του ευρετηρίου CGI περιλαμβάνει έναν πολύ μεγάλο αριθμό συνδυασμών γραφημάτων, τότε προτείνεται η διάσπασή του με συγκεκριμένο τρόπο [58, 78].

Στο ίδιο πνεύμα με τη δομή αποθήκευσης του συστήματος G^* , το ευρετήριο CGI μπορεί επίσης να χρησιμοποιηθεί για την διαδικασία απάντησης ερωτημάτων, έτσι ώστε να εξασφαλίζεται ότι κάθε εκδοχή μιας κορυφής ή ακμής εξετάζεται μόνο μία φορά κατά την απάντηση ενός ερωτήματος.

Επιπλέον, το σύστημα G^* παρέχει τρεις βασικούς τύπους που μπορούν να χρησιμοποιηθούν για την δημιουργία τελεστών διαχείρισης ερωτημάτων σε γραφήματα: περιλήψεις (summaries), συνδυαστές (combiners), και τελεστές μαζικά σύγχρονου παραλληλισμού (bulk synchronous parallel operators). Τέλος, η εργασία [59] μελετά την αναπαραγωγή στιγμιότυπων και τεχνικές διανομής τους.

Δημοσίευση	Περιβάλλον	Σκοπός/Προσέγγιση
[73]	Κεντριοποιημένο	Αποθήκευση & Ανάκτηση Στιγμιοτύπων, Συντομότερες Διαδρομές, Ερωτήματα Κεντρικής Εγγύτητας
[55]	Κεντριοποιημένο	Αποθήκευση & Ανάκτηση Στιγμιοτύπων, Διαδικασία Ερωτημάτων Δύο-Φάσεων
[15]	Κεντριοποιημένο	Αποθήκευση & Ανάκτηση Στιγμιοτύπων, Συμπαγείς και Αυτο-Δεικτοδοτούμενες Δομές Δεδομένων
[2]	Κεντριοποιημένο	Ιστορικά Ερωτήματα Απόστασης
[52]	Κεντριοποιημένο	Ερωτήματα Συντομότερων Διαδρομών
[87]	Κεντριοποιημένο	Ανακάλυψη των Συχνότερα Μεταβαλλόμενων Συνιστωσών
[76]	Κεντριοποιημένο	Ιστορικά Ερωτήματα Προσπελασιμότητας
[54]	Κατανεμημένο	Αποθήκευση & Ανάκτηση Στιγμιοτύπων
[58]	Κατανεμημένο	Αποθήκευση & Ανάκτηση Στιγμιοτύπων

Πίνακας 3.1: Σύνοψη των Σύγχρονων Γραφοκεντρικών Μοντέλων της Ενότητας 3.3.

3.4 Αποτίμηση Μοντέλων

Ένα μεγάλο πλήθος σύγχρονων (δυναμικά μεταβαλλόμενων) δικτύων μπορούν να μοντελοποιηθούν ως δομές γραφημάτων οι οποίες είναι δυναμικές (χρονικά μεταβαλλόμενες) και συνεχώς εξελισσόμενες.

Στην Ενότητα 3.3 παρουσιάστηκε ένα αντιπροσωπευτικό δείγμα σύγχρονων κεντριοποιημένων και αποκεντριοποιημένων γραφοκεντρικών δυναμικών μοντέλων, με τα σημαντικότερα χαρακτηριστικά τους, τα οποία σε συνοπτική μορφή παρουσιάζονται στον Πίνακα 3.1. Ανάλογα με τις απαιτήσεις και τις ιδιαιτερότητες κάθε δικτύου και εφαρμογής, αλλά και σε συνδυασμό με την πολυπλοκότητα των προβλημάτων που δίνονται προς επίλυση, χρησιμοποιείται το καταλληλότερο κατά περίπτωση μοντέλο.

Η μελέτη της βιβλιογραφίας που αφορά τα σύγχρονα δυναμικά γραφοκεντρικά μοντέλα (Ενότητα 3.3) έγινε με στόχο τη διαπίστωση του κατά πόσο τα μοντέλα αυτά μπορούν να μοντελοποιήσουν με αποδοτικό τρόπο συστήματα πληροφορίας πινάκων δρομολογίων που αφορούν MMM. Τα μοντέλα της Ενό-

τητας 3.3 προσφέρουν εργαλεία απάντησης ερωτημάτων και διαχείρισης δυναμικά μεταβαλλόμενων ή εξελισσόμενων γραφημάτων, τα οποία μπορούν να μοντελοποιήσουν συστήματα πληροφορίας πινάκων δρομολογίων καθώς και καθυστερήσεων. Τα εργαλεία όμως αυτά δεν είναι τόσο αποδοτικά όσον αφορά την απαίτηση για γρήγορη επεξεργασία (σε πραγματικό χρόνο) ενός τεράστιου πλήθους ερωτημάτων NA (ή EAM), καθώς και την ταχεία ενημέρωση της πληροφορίας σε περίπτωση καθυστερήσεων των MMM, σε πίνακες δρομολογίων με εκατομύρια αναχωρήσεις/αφίξεις και στοιχειώδεις συνδέσεις.

Ειδικά για την ενημέρωση της πληροφορίας σε περίπτωση μεταβολών (πχ σε περίπτωση καθυστερήσεων των MMM), μια πρόσφατη πειραματική μελέτη [57] στο μοντέλο των δυναμικά μεταβαλλόμενων ή εξελισσόμενων γραφημάτων κατέδειξε χρόνους ενημέρωσης μερικών δευτερολέπτων, ενώ — όπως θα δούμε στο Κεφάλαιο 5 — επεκτάσεις και νέες μοντελοποιήσεις βασισμένες στο ρεαλιστικό χρονοεκτεταμένο μοντέλο (οι οποίες θα παρουσιαστούν στο Κεφάλαιο 4) επιτυγχάνουν χρόνους ενημέρωσης της πληροφορίας πινάκων δρομολογίων σε περίπτωση καθυστερήσεων σε χρόνο λιγότερο των 0.3 msec. Επίσης, η διατήρηση ενός πίνακα δρομολογίων ως μια δυναμικά εξελισσόμενη ακολουθία γραφημάτων αυξάνει πολύ την πολυπλοκότητα αναπαράστασης και δαπανά πολύ περισσότερο χώρο σε σχέση με το ρεαλιστικό χρονοεκτεταμένο μοντέλο (Ενότητα 3.2).

Από τα παραπάνω προκύπτει ότι τα σύγχρονα δυναμικά γραφοκεντρικά μοντέλα (Ενότητα 3.3) δεν είναι τόσο αποδοτικά για την μοντελοποίηση πληροφοριακών συστημάτων πινάκων δρομολογίων MMM και καθυστερήσεων σε αυτά. Για αυτό το λόγο, η έρευνα της παρούσας διδακτορικής διατριβής προσανατολίστηκε γύρω από το υπάρχον χρονοεκτεταμένο μοντέλο, που αποδεδειγμένα είναι κατάλληλο για την μοντελοποίηση πληροφορίας πινάκων δρομολογίων. Ο στόχος ήταν να διερευνηθεί το κατά πόσο μπορεί το μοντέλο αυτό να επεκταθεί έτσι ώστε να ανταποκρίνεται στις αυξημένες απαιτήσεις των πραγματικών εφαρμογών.

Στο επόμενο κεφάλαιο παρουσιάζονται αναλυτικά μια επέκταση του ρεαλιστικού χρονοεκτεταμένου μοντέλου καθώς και η εισαγωγή ενός νέου χρονοεκτεταμένου μοντέλου, μαζί με τους αλγορίθμους απάντησης ερωτημάτων NA και EAM, καθώς και με τους αλγορίθμους ενημέρωσης της δομής σε περίπτωση

καθυστερήσεων.

Κεφάλαιο 4

Νέα Γραφοκεντρικά Μοντέλα Αναπαράστασης Πληροφορίας Δρομολογίων

Στο κεφάλαιο αυτό παρουσιάζονται τα δύο νέα γραφοκεντρικά μοντέλα αναπαράστασης πινάκων δρομολογίων που σχεδιάστηκαν και αναπτύχθηκαν στα πλαίσια της παρούσας διδακτορικής διατριβής. Αρχικά παρουσιάζεται το α-νηγμένο ρεαλιστικό χρονοεκτεταμένο μοντέλο που ουσιαστικά αποτελεί μία (απλοποιημένη) έκδοση του αυθεντικού ρεαλιστικού μοντέλου, χωρίς όμως να υστερεί στην ολοκληρωμένη αποτύπωση των πληροφοριών ενός πίνακα δρομολογίων. Στη συνέχεια, παρουσιάζεται το δυναμικό μοντέλο πινάκων δρομολογίων το οποίο διαφέρει από το αυθεντικό ρεαλιστικό χρονοεκτεταμένο μοντέλο. Επιπλέον και για τα δύο νέα μοντέλα, παρουσιάζονται αλγόριθμοι επίλυσης των βασικών προβλημάτων NA και EAM, καθώς και αλγόριθμοι για την περίπτωση καθυστερήσεων. Όπως κατέδειξαν και τα πειραματικά αποτελέσματα (Κεφάλαιο 5), τα δύο νέα μοντέλα βελτιώνουν σημαντικά το χρόνο απόκρισης ερωτημάτων σε σχέση με το αυθεντικό ρεαλιστικό μοντέλο.

4.1 Ανηγμένο Ρεαλιστικό Χρονοεκτεταμένο Μοντέλο

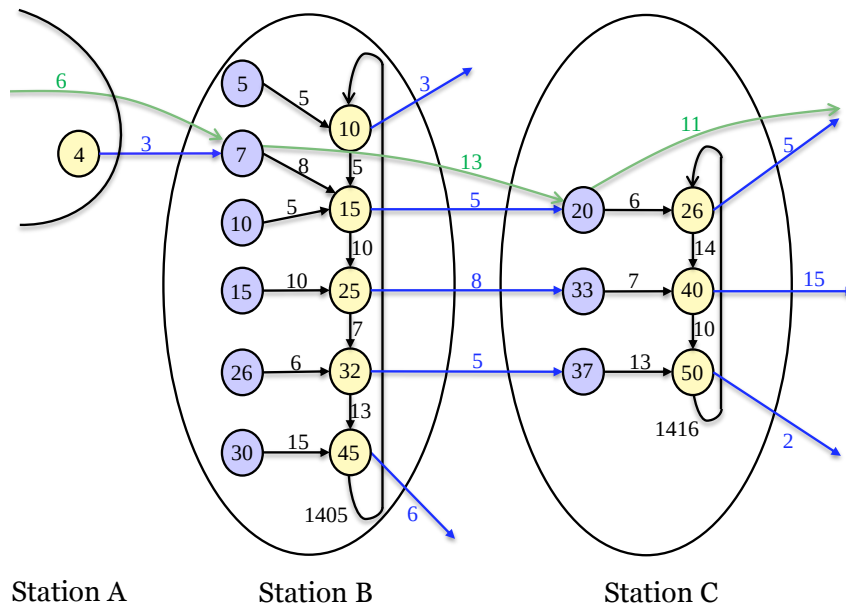
Σε αυτή την ενότητα παρουσιάζεται μια απλοποιημένη παραλλαγή του ρεαλιστικού χρονοεκτεταμένου μοντέλου, η οποία ονομάζεται *ανηγμένο* (ρεαλιστικό) χρονοεκτεταμένο μοντέλο. Βασικός στόχος της δημιουργίας του είναι η μείωση του μεγέθους του ρεαλιστικού χρονοεκτεταμένου γραφήματος. Για το μοντέλο αυτό, δίνεται η κατασκευή του γραφήματος καθώς και οι αλγόριθμοι απόκρισης ερωτημάτων και ενημέρωσης της πληροφορίας πινάκων δρομολογίων στην περίπτωση καθυστερήσεων.

4.1.1 Το Μοντέλο

Η βασική διαφορά μεταξύ του ανηγμένου και του αυθεντικού ρεαλιστικού χρονοεκτεταμένου μοντέλου είναι ότι στο ανηγμένο τόσο οι κορυφές μετεπιβίβασης όσο και οι ακμές μετεπιβίβασης-αναχώρησης διαγράφονται από το γράφημα, με κατάλληλη ενημέρωση του κόστους κάποιων ακμών, έτσι ώστε να μη χάνεται καμία πληροφορία δρομολογίου.

Πιο συγκεκριμένα, οι κορυφές αναχώρησης συνενώνονται με τις αντίστοιχες κορυφές μετεπιβίβασης, οι οποίες (κορυφές μετεπιβίβασης) καταργούνται, και οι κορυφές άφιξης συνδέονται απευθείας με τις κορυφές αναχώρησης με ακμές των οποίων το κόστος ισούται με τη χρονική διαφορά των κορυφών που συνδέει. Επίσης, αν υπάρχει μία ακμή άφιξης-αναχώρησης σε ένα σταθμό s_1 και η αναχώρηση (από τον s_1) αντιστοιχεί σε άφιξη σε ένα σταθμό s_2 στο αρχικό ρεαλιστικό χρονοεκτεταμένο γράφημα, τότε αυτή η ακμή μετατρέπεται σε απευθείας σύνδεση από την κορυφή άφιξης στον s_1 στην κορυφή άφιξης στον s_2 στο ανηγμένο χρονοεκτεταμένο μοντέλο.

Το Σχήμα 4.1 δείχνει το ανηγμένο χρονοεκτεταμένο γράφημα που αντιστοιχεί στο ρεαλιστικό χρονοεκτεταμένο γράφημα του Σχήματος 3.1. Οι κορυφές μετεπιβίβασης (γκρί) και οι ακμές μετεπιβίβασης-αναχώρησης (από γκρι σε κίτρινη κορυφή) του Σχήματος 3.1 δεν υπάρχουν στο γράφημα του Σχήματος 4.1. Οι κορυφές άφιξης (μπλέ) συνδέονται απευθείας με τις κορυφές αναχώρησης (κίτρινες) στον ίδιο σταθμό. Μια ακμή άφιξης-αναχώρησης στον ίδιο σταθμό



Σχήμα 4.1: Το γράφημα του ανηγμένου χρονοεκτεταμένου μοντέλου που αντιστοιχεί στο χρονοεκτεταμένο γράφημα του Σχήματος 3.1. Οι κορυφές άφιξης απεικονίζονται με μπλε χρώμα, ενώ οι κορυφές αναχώρησης, οι οποίες είναι διατεταγμένες με βάση το χρόνο αναχώρησης, απεικονίζονται με κίτρινο χρώμα. Στον ίδιο σταθμό, οι κορυφές άφιξης συνδέονται απευθείας με τις κορυφές αναχώρησης.

μετατρέπεται σε ακμή σύνδεσης μεταξύ κορυφών άφιξης γειτονικών σταθμών — π.χ. η πράσινη ακμή με άκρα τις κορυφές 7 και 15 στο Σχήμα 3.1 μετατρέπεται στην πράσινη ακμή με άκρα τις κορυφές 7 και 20 στο Σχήμα 4.1.

Το παρακάτω θεώρημα παρουσιάζει τις απαιτήσεις χώρου του ανηγμένου χρονοεκτεταμένου γραφήματος.

Θεώρημα 4 *Το ανηγμένο χρονοεκτεταμένο γράφημα $G = (V, E)$ που αναπαριστάνει έναν πίνακα δρομολογίων $\mathcal{T} = (\mathcal{Z}, \mathcal{B}, \mathcal{C})$ έχει $n = |V| = 2|\mathcal{C}|$ κορυφές, $3|\mathcal{C}| \leq m = |E| \leq 4|\mathcal{C}|$ ακμές, και απαιτεί χώρο $O(|\mathcal{C}|)$.*

Απόδειξη. Η απόδειξη προκύπτει από εκείνη του Θεωρήματος 1 και το γεγονός ότι δεν υπάρχουν πλέον οι $|\mathcal{C}|$ κορυφές μετεπιβίβασης και οι μεταξύ τους $|\mathcal{C}|$ ακμές. ■

Το ανηγμένο χρονοεκτεταμένο γράφημα έχει σαν αποτέλεσμα την μείωση του μεγέθους του αρχικού ρεαλιστικού χρονοεκτεταμένου γραφήματος κατά

$|\mathcal{C}|$ κορυφές και $|\mathcal{C}|$ ακμές, γεγονός που συνεπάγεται και μικρότερο χρόνο εξερεύνησης του νέου γραφήματος.

4.1.2 Αλγόριθμος Απάντησης Ερωτημάτων

Σε αυτή την ενότητα παρουσιάζεται μία παραλλαγή του αλγορίθμου Dijkstra, κατάλληλα προσαρμοσμένη στο χρονοεκτεταμένο γράφημα, η οποία ονομάζεται TE-QH. Η παραλλαγή αυτή εκμεταλλεύεται την δομή και τους περιορισμούς του ανηγμένου χρονοεκτεταμένου μοντέλου με αποτέλεσμα να βελτιώνεται ο χρόνος απόκρισης των ερωτημάτων. Αξίζει να σημειωθεί ότι, δεν επηρεάζεται η συμπεριφορά χειρότερης περίπτωσης του αλγορίθμου, η οποία παραμένει $O(|\mathcal{C}| \log |\mathcal{C}|)$. Ωστόσο, στο Κεφάλαιο 5, όπου παρουσιάζεται εκτενής πειραματική αξιολόγηση των αλγορίθμων, διαπιστώνεται ότι η απόδοση του TE-QH στην πράξη είναι πολύ καλύτερη.

Τα βασικά σημεία του αλγορίθμου TE-QH είναι τα εξής.

Πρώτον, με σκοπό την μείωση του μέγεθους της ουράς προτεραιότητας PQ του αλγορίθμου Dijkstra (Ενότητα 2.3.1) εισάγονται στην PQ μόνο οι κορυφές άφιξης. Αυτή η επιλογή μπορεί να αποδειχθεί αρκετά επωφελής γιατί το υπολογιστικό κόστος του αλγορίθμου εξαρτάται, σε πολύ μεγάλο βαθμό, από το πλήθος των λειτουργιών που εκτελούνται στην ουρά προτεραιότητας.

Όμως, για να διασφαλιστεί η ορθότητα του αλγορίθμου σε σχέση με τον αυθεντικό αλγόριθμο Dijkstra, επεκτείνεται το βήμα της χαλάρωσης μόνο για την περίπτωση των κορυφών αναχώρησης. Δηλαδή, για κάθε χαλάρωση ακμής (u, v) , για την οποία η κορυφή v είναι κορυφή αναχώρησης και η απόστασή της έχει μειωθεί, συνεχίζεται η διαδικασία της χαλάρωσης και στις εξερχόμενες ακμές αυτής της κορυφής v . Σε αυτή την περίπτωση, το βάθος της χαλάρωσης των ακμών είναι άνω φραγμένο από τον αριθμό των κορυφών αναχώρησης που ανήκουν σε μια κατάλληλα ορισμένη ακυκλική συνιστώσα μεταξύ γειτονικών σταθμών. Αυτή η ακυκλική συνιστώσα, μεταξύ δύο γειτονικών σταθμών s_A και s_B , αποτελείται από όλες τις απλές διαδρομές $(v_1, v_2, \dots, v_{k-1}, v_k)$, όπου v_1 είναι μία κορυφή άφιξης της s_A , οι κορυφές $v_2, \dots, v_{k-1}$ είναι κορυφές αναχώρησης της s_A , και η v_k είναι μία κορυφή άφιξης του γειτονικού σταθμού s_B . Για παράδειγμα, στο Σχήμα 4.1, το ακυκλικό υπογράφημα, μεταξύ των σταθμών

B και C, επάγεται από την κορυφή άφιξης 7 και τις κορυφές αναχώρησης 15, 25, 32 στο σταθμό B, τις κορυφές άφιξης 20, 33, 37 στον σταθμό C, και από τις αντίστοιχες διαδρομές (7,15,20), (7,15,25,33) και (7,15,25,32,37).

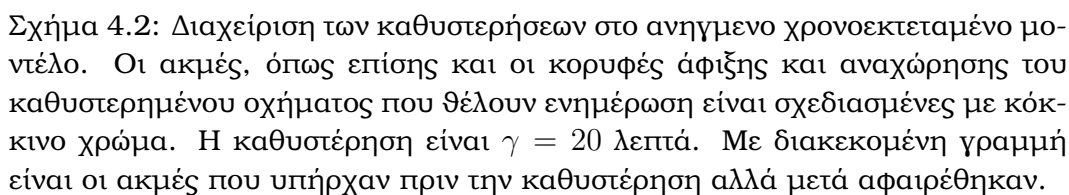
Δεύτερον, για τον περιορισμό του χώρου αναζήτησης, τροποποιούνται τα κριτήρια εξερεύνησης κορυφών ώστε να είναι πιο επιλεκτικά. Πιο συγκεκριμένα, παίρνοντας ως αναφορά τον αρχικό σταθμό s_S , ενώ εκτελείται ο αλγόριθμος Dijkstra, παρακολουθείται ο χρόνος νωρίτερη άφιξης στους τρέχοντες σταθμούς τους οποίους επισκέπτεται ο αλγόριθμος. Γενικά, μεταξύ δύο γειτονικών συνδεδεμένων σταθμών s_A και s_B μπορεί να υπάρχουν πολλαπλές διαδρομές με διαφορετικούς χρόνους αναχώρησης από τον s_A . Στόχος είναι ο αποκλεισμός από την αναζήτηση του αλγόριθμου, των σταθμών που δεν αποτελούν μέρος της βέλτιστης διαδρομής. Για να επιτευχθεί αυτό, αξιοποιείται το γεγονός ότι οι κορυφές αναχώρησης ταξινομούνται σε αύξουσα διάταξη ως προς την συσχετιζόμενη χρονοσφραγίδα τους (χρόνος αναχώρησης). Επομένως, σε ένα συγκεκριμένο βήμα της εκτέλεσης του αλγορίθμου απάντησης ερωτημάτων, μπορεί να παραληφθούν (δηλαδή να εξαιρεθούν από την αναζήτηση) όλες οι κορυφές αναχώρησης των οποίων ο χρόνος αναχώρησης είναι μεγαλύτερος ή ίσος με τον τρέχοντα ελάχιστο χρόνο άφιξης στον s_B συν (επειδή είμαστε στην περίπτωση του ρεαλιστικού χρονοεκτεταμένου μοντέλου) το χρόνο μετεπιβίβασης $transfer(s_B)$ του σταθμού s_B . Είναι προφανές ότι οι κορυφές αναχώρησης που έχουν παραληφθεί δεν μπορούν να δώσουν βέλτιστη λύση σε ερωτήματα εύρεσης νωρίτερης άφιξης.

Η συζήτηση σε αυτή την ενότητα συνοψίζεται στο παρακάτω θεώρημα.

Θεώρημα 5 *Ο αλγόριθμος TE-QH επιλύει τα προβλήματα NA και EAM σε χρόνο $O(|C| \log |C|)$ στο ανηγμένο χρονοεκτεταμένο γραφοκεντρικό μοντέλο.*

Σημειώνεται ότι ο αλγόριθμος TE-QH που μόλις περιγράφηκε μπορεί να χρησιμοποιηθεί σε συνδυασμό με τις υπάρχουσες ευρετικές τεχνικές επιτάχυνσης, κυρίως με την ALT (Ενότητα 2.4). Στο Κεφάλαιο 5 (πειραματική αξιολόγηση) θα καταδειχθεί πως αυτός ο συνδυασμός προσφέρει σημαντική βελτίωση στην απόδοση του αλγορίθμου.

Για την ενημέρωση του ανηγμένου χρονοεκτεταμένου γράφηματος στην περίπτωση που συμβεί μία καθυστέρηση σε ένα όχημα, σχεδιάστηκε και υλοποιήθηκε ο αλγόριθμος TE-UH. Να σημειωθεί ότι, αν και δεν βελτιώνεται ο ασυμπτωτικός χρόνος χειρότερης περίπτωσης σε σχέση με τον αλγόριθμο TE-U, ο νέος αλγόριθμος TE-UH διαχείρισης καθυστερήσεων στο ανηγμένο χρονοεκτεταμένο γράφημα είναι σημαντικά καλύτερος στην πράξη.



Έστω ότι ένα όχημα φτάνει στο σταθμό B με καθυστέρηση $\gamma = 20$ λεπτών. Αρχικά, αυξάνονται κατά γ λεπτά τα κόστη των εισερχόμενων ακμών της κορυφής άφιξης στο σταθμό B , η οποία αντιπροσωπεύει το καθυστερημένο όχημα

(εισερχόμενες ακμές της κορυφής άφιξης 7 στο Σχήμα 4.1 πριν την καθυστέρηση, και εισερχόμενες ακμές της κορυφής άφιξης 27 στο Σχήμα 4.2 μετά την καθυστέρηση).

Δεύτερον, για κάθε σταθμό από τον οποίο διέρχεται το καθυστερημένο όχημα, ενημερώνονται οι ακμές άφιξης-αναχώρησης και αναχώρησης-αναχώρησης οι οποίες επηρεάζονται. Αυτό πραγματοποιείται με την αλλαγή των βαρών τους και με την επανατοποθέτησή τους όπου απαιτείται. Σε κάθε τέτοιο σταθμό, εισάγεται ένα ζεύγος μιας καθυστερημένης κορυφής x (η κορυφή άφιξης 7 στο Σχήμα 4.1 και η καθυστερημένη κορυφή άφιξης 27 στο Σχήμα 4.2) και μιας καθυστερημένης κορυφής αναχώρησης y (η κορυφή αναχώρησης 15 στο Σχήμα 4.1 και η καθυστερημένη κορυφή αναχώρησης 35 στο Σχήμα 4.2). Ανάλογα με το μέγεθος της καθυστέρησης, μπορεί να υπάρχει τουλάχιστον μία κορυφή αναχώρησης η οποία θα πρέπει να συνδεθεί με μία νέα νωρίτερη χρονικά κορυφή αναχώρησης. Αυτό το γεγονός απαιτεί μία τροποποίηση της τοπολογίας του ανηγμένου χρονοεκτεταμένου γραφήματος, ώστε να συνεχίζει να είναι έγκυρο.

Οι ακμές οι οποίες επηρεάζονται από την καθυστέρηση βρίσκονται αποκλειστικά μέσα στους σταθμούς, και συγκεκριμένα είναι εκείνες οι ακμές που ικανοποιούν μία από τις παρακάτω συνθήκες:

- Η ουρά (αρχική κορυφή) τους είναι η καθυστερημένη κορυφή άφιξης x (η ακμή από την κορυφή άφιξης 7 στην κορυφή αναχώρησης 15 στο Σχήμα 4.1, και η ακμή από την κορυφή άφιξης 27 στην κορυφή αναχώρησης 32 στο Σχήμα 4.2).
- Η κεφαλή (τελική κορυφή) τους είναι η καθυστερημένη κορυφή αναχώρησης y , μέσω της οποίας το όχημα συνεχίζει το ταξίδι του σε επόμενο σταθμό (η ακμή από την κορυφή άφιξης 10 στην κορυφή αναχώρησης 15 στο Σχήμα 4.1, και η ακμή από την κορυφή άφιξης 10 στην κορυφή αναχώρησης 25 στο Σχήμα 4.2).
- Η κεφαλή τους είναι η επόμενη (χρονικά) κορυφή από την κορυφή αναχώρησης y (η ακμή από την κορυφή άφιξης 30 στην κορυφή αναχώρησης 45 στο Σχήμα 4.1, και η ακμή από την κορυφή άφιξης 30 στην κορυφή

αναχώρησης 35 στο Σχήμα 4.2).

Για να διατηρηθεί η ιδιότητα της ταξινόμησης των κορυφών αναχώρησης σύμφωνα με τη χρονοσφραγίδα τους, πρέπει να μετακινηθεί η καθυστερημένη κορυφή αναχώρησης στην σωστή της θέση. Αυτό γίνεται με τρόπο παρόμοιο με αυτόν της μετακίνησης ενός στοιχείου από μία θέση σε μία άλλη σε μια (ταξινομημένη) συνδεδεμένη λίστα. Στη συνέχεια, το μόνο που χρειάζεται είναι η σύνδεση των κορυφών άφιξης που έχουν επηρεαστεί από την καθυστέρηση με τις κατάλληλες κορυφές αναχώρησης, έτσι ώστε να τηρείται ο χρόνος μετα-πιθίδασης μέσα στο σταθμό. Παράλληλα, ενημερώνονται οι ακμές με τα νέα σωστά κόστη. Οι παραπάνω ενέργειες απαιτούν μόνο αλλαγή των δεικτών των κορυφών των ακμών και των κοστών τους, το οποίο ελαχιστοποιεί τον χρόνο ενημέρωσης, και επίσης, σε αντίθεση με το αυθεντικό ρεαλιστικό μοντέλο, το ανηγμένο ρεαλιστικό μοντέλο διατηρεί τον αριθμό των ακμών (που χρειάζονται ενημέρωση) σταθερό.

Στο Σχήμα 4.2 παρουσιάζεται το αποτέλεσμα της εκτέλεσης του αλγορίθμου ενημέρωσης στο ανηγμένο χρονοεκτεταμένο γράφημα του Σχήματος 4.1, στην περίπτωση του παραδείγματος μας (που υποθέτει καθυστέρηση 20 λεπτών σε ένα όχημα).

Η συζήτηση σε αυτή την ενότητα συνοψίζεται στο παρακάτω θεώρημα.

Θεώρημα 6 *Ο αλγόριθμος TE-UH απαιτεί χρόνο $O(|C| \log |C|)$ για να ενημερώσει έναν πίνακα δρομολογίων που αναπαριστάνεται από το ανηγμένο χρονοεκτεταμένο γραφοκεντρικό μοντέλο μετά από μια καθυστέρηση οχήματος.*

Το μοναδικό μειονέκτημα αυτής της προσέγγισης είναι ότι οι κορυφές αναχώρησης μπορεί πλέον να μην μπορούν να ταξινομηθούν κατά βέλτιστο τρόπο μέσα στις πλοκάδες μνήμης και επομένως να χειροτερεύουν την τοπικότητα των αναφορών. Αυτό μπορεί να έχει αντίκτυπο στη χειροτέρευση του χρόνου εκτέλεσης του αλγορίθμου απάντησης ερωτημάτων. Προκειμένου να μειωθεί αυτή η συνέπεια, όλες οι κορυφές αναχώρησης σε κάθε σταθμό ομαδοποιούνται σε μια αρχική φάση (με συμπιεσμένο τρόπο) στη μνήμη.

Τα αρχικά πειράματα που πραγματοποιήθηκαν έδειξαν ότι ο αλγόριθμος TE-UH είναι τουλάχιστον 50% ταχύτερος από τον TE-U.

4.2 Το Δυναμικό Μοντέλο Πίνακα Δρομολογίων

Σε αυτή την ενότητα, παρουσιάζεται η νέα γραφοκεντρική προσέγγιση, που ονομάζεται *δυναμικό μοντέλο πίνακα δρομολογίων* (dynamic timetable model — DTM). Επιπλέον, παρουσιάζονται και οι κατάλληλοι αλγόριθμοι απάντησης ερωτημάτων για την επίλυση των προβλημάτων NA και EAM καθώς και ο αλγόριθμος ενημέρωσης στην περίπτωση καθυστερήσεων.

4.2.1 Το Μοντέλο

Για ένα πίνακα δρομολογίων $\mathcal{T} = (\mathcal{Z}, \mathcal{B}, \mathcal{C})$, ορίζεται ένα κατευθυνόμενο γράφημα $G = (V, E)$ που ονομάζεται *δυναμικό γράφημα πινάκων δρομολογίων* (dynamic timetable graph και εν συντομία *γράφημα DTM*) και μία συνάρτηση κόστους $wt : E \rightarrow \mathbb{N}$, ως εξής:

- Για κάθε σταθμό S του $\mathcal{B}$, δημιουργείται μία κορυφή s_S , η οποία ονομάζεται *κορυφή μεταγωγής* του σταθμού S , και προστίθεται στο σύνολο V .
- Για κάθε στοιχειώδη σύνδεση $c = (Z, S_d, S_a, t_d, t_a) \in \mathcal{C}$, προστίθεται στο σύνολο κορυφών V μια κορυφή d_c , η οποία ονομάζεται *κορυφή αναχώρησης* της c , ενώ προστίθεται στο σύνολο των ακμών E μία ακμή (d_c, s_{S_a}) , η οποία ονομάζεται *ακμή σύνδεσης* της c και η οποία συνδέει την κορυφή d_c με την κορυφή μεταγωγέα s_{S_a} του S_a .
- Για κάθε στοιχειώδη σύνδεση $c = (Z, S_d, S_a, t_d, t_a) \in \mathcal{C}$, προστίθεται στο σύνολο ακμών E μία ακμή (s_{S_d}, d_c) , η οποία ονομάζεται *ακμή μεταγωγής* και η οποία συνδέει την κορυφή μεταγωγής s_{S_d} του σταθμού αναχώρησης S_d με την κορυφή αναχώρησης d_c της c .
- Για κάθε όχημα $Z \in \mathcal{Z}$ το οποίο ταξιδεύει κατά μήκος της διαδρομής $(c_1, c_2, \dots, c_k)$, προστίθεται στο σύνολο ακμών E μία ακμή $(d_{c_i}, d_{c_{i+1}})$, η οποία ονομάζεται *ακμή οχήματος* και η οποία συνδέει την κορυφή αναχώρησης d_{c_i} της c_i με την κορυφή αναχώρησης $d_{c_{i+1}}$ της c_{i+1} , για κάθε $i = 1, 2, \dots, k - 1$.

- Το κόστος κάθε ακμής σύνδεσης (d_c, s_{S_a}) ορίζεται ως $wt(d_c, s_{S_a}) = \Delta(t_a(c), t_d(c))$.
- Το κόστος κάθε ακμής οχήματος $(d_{c_i}, d_{c_{i+1}})$ ορίζεται ως $wt(d_{c_i}, d_{c_{i+1}}) = \Delta(t_d(c_i), t_d(c_{i+1}))$.
- Το κόστος κάθε ακμής μεταγωγής ορίζεται αρχικά ως $(s_{S_d}, d_c) = +\infty$.

Για κάθε σταθμό, οι κορυφές αναχώρησης d_c διατάσσονται κατά αύξοντα χρόνο άφιξης στον αμέσως επόμενο σταθμό S_a , δηλαδή διατάσσονται κατά την τιμή $t_d(c) + wt(d_c, s_{S_a})$. Επιπλέον, για κάθε κορυφή μεταγωγής s_S , αποθηκεύεται ο σταθμός S με τον οποίο συσχετίζεται, ενώ για κάθε κορυφή αναχώρησης d_c διατηρείται τόσο ο συσχετιζόμενος χρόνος αναχώρησης $t_d(c)$ όσο και το όχημα $Z(c)$ της συσχετιζόμενης σύνδεσης c .

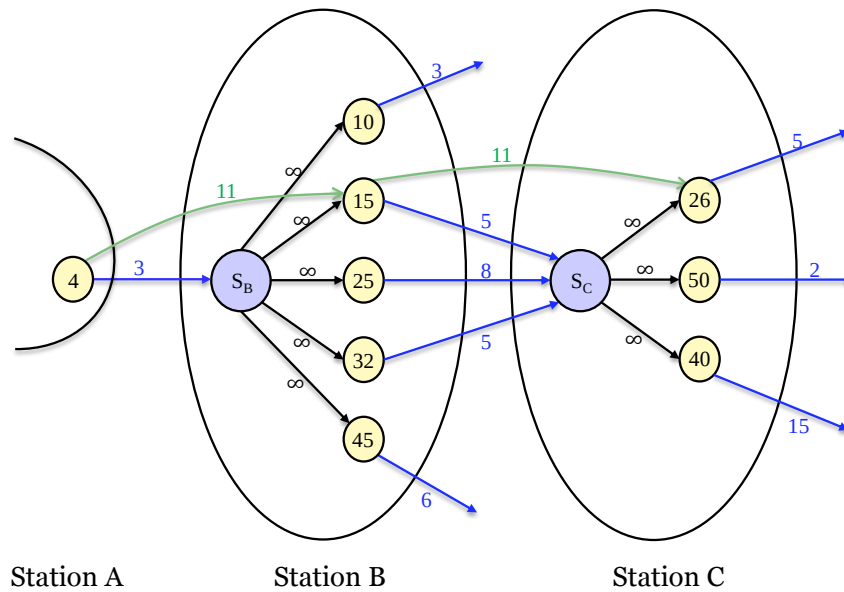
Το Σχήμα 4.3 απεικονίζει το γράφημα DTM που αντιστοιχεί στο ρεαλιστικό χρονοεκτεταμένο γράφημα του Σχήματος 3.1.

Το παρακάτω θεώρημα παρουσιάζει τις απαιτήσεις χώρου του γραφήματος DTM.

Θεώρημα 7 Το γράφημα DTM $G = (V, E)$ που αναπαριστάνει έναν πίνακα δρομολογίων $\mathcal{T} = (\mathcal{Z}, \mathcal{B}, \mathcal{C})$ έχει $n = |V| = |\mathcal{B}| + |\mathcal{C}|$ κορυφές, $m = |E| \leq 3|\mathcal{C}|$ ακμές, και απαιτεί χώρο $O(|\mathcal{C}|)$.

Απόδειξη. Το γράφημα DTM G έχει, από την κατασκευή του, ένα πλήθος $n = |\mathcal{B}| + |\mathcal{C}|$ κορυφών και ένα πλήθος ακμών $m \leq 3|\mathcal{C}|$. Ο τελευταίος ισχυρισμός ισχύει επειδή οι ακμές αντιστοιχούν σε στοιχειώδεις συνδέσεις, οι οποίες συνδέουν κορυφές αναχώρησης με κορυφή μεταγωγής σε γειτονικό σταθμό (μπλε ακμές στο Σχήμα 4.3), κορυφή μεταγωγής με κορυφές αναχώρησης στον ίδιο σταθμό (μαύρες ακμές στο Σχήμα 4.3), και κορυφή αναχώρησης σε έναν σταθμό με κορυφή αναχώρησης σε γειτονικό σταθμό (πράσινες ακμές στο Σχήμα 4.3). Οι τελευταίες ακμές μπορεί να μην περιλαμβάνουν όλες τις κορυφές αναχώρησης.

Η αποθήκευση της επιπλέον πληροφορίας απαιτεί χώρο $O(|\mathcal{B}|)$ για την αποθήκευση του συσχετιζόμενου σταθμού με κάθε κορυφή μεταγωγής, και $O(|\mathcal{C}|)$



Σχήμα 4.3: Το γράφημα DTM που αντιστοιχεί στο ρεαλιστικό χρονοεκτεταμένο γράφημα του Σχήματος 3.1. Οι κορυφές μεταγωγής απεικονίζονται με μπλε χρώμα, ενώ οι κορυφές αναχώρησης, που είναι διατεταγμένες με αύξοντα χρόνο άφιξης στον αμέσως επόμενο σταθμό, απεικονίζονται με κίτρινο χρώμα. Μέσα σε κάθε κορυφή αναχώρησης καταγράφεται ο χρόνος αναχώρησης της αντίστοιχης στοιχειώδους σύνδεσης. Οι ακμές σύνδεσης απεικονίζονται με μπλέ χρώμα, οι ακμές μεταγωγής με μαύρο, ενώ οι ακμές οχήματος απεικονίζονται με πράσινο.

χώρο για την αποθήκευση του συσχετιζόμενου χρόνου αναχώρησης και οχήματος με κάθε κορυφή αναχώρησης. Το θεώρημα προκύπτει από το γεγονός ότι $|\mathcal{C}| \geq \max\{|\mathcal{B}|, |\mathcal{Z}|\}$. ■

4.2.2 Αλγόριθμος Απάντησης Ερωτημάτων

Στην ενότητα αυτή παρουσιάζεται ο αλγόριθμος απάντησης ερωτημάτων, που ονομάζεται DTM-Q, και είναι κατάλληλος για την επίλυση τόσο του προβλήματος νωρίτερης άφιξης (NA) όσο και του προβλήματος εύρεσης του ελάχιστου αριθμού μετεπιβιβάσεων (EAM) στο γράφημα DTM. Αρχικά περιγράφεται η περίπτωση επίλυσης ερωτημάτων NA και στη συνέχεια παρουσιάζεται η προσαρμογή του αλγορίθμου ώστε να επιλύει ερωτήματα για εύρεση διαδρομών EAM.

Δεδομένου ενός γραφήματος DTM G , η απάντηση ενός ερωτήματος NA (S, T, t_S) πραγματοποιείται με την εκτέλεση μιας παραλλαγής του αλγορίθμου Dijkstra στο G , ξεκινώντας από την κορυφή μεταγωγής s_S του σταθμού S , η οποία περιγράφεται στο υπόλοιπο αυτής της ενότητας.

Πριν την αναλυτική παρουσίαση του αλγορίθμου DTM-Q, περιγράφεται μία επιπλέον δομή δεδομένων που χρησιμοποιείται από τον αλγόριθμο σε σχέση με τον κλασικό αλγόριθμο του Dijkstra.

Συγκεκριμένα, για κάθε κορυφή μεταγωγής s_A του σταθμού A , ο αλγόριθμος DTM-Q αποθηκεύει:

- Ένα διάνυσμα λογικών τιμών (boolean flags) D_A , το μέγεθος του οποίου προκύπτει από το αριθμό των σταθμών A' για τους οποίους υπάρχει τουλάχιστον μία στοιχειώδης σύνδεση που αναχωρεί από τον σταθμό A και φτάνει στον σταθμό A' . Το στοιχείο του D_A που συσχετίζεται με τον σταθμό A' ως δηλώνεται $D_A[A']$. Αρχικά, σε όλες τα στοιχεία του D_A εκχωρείται η τιμή ψεύδους.
- Μαζί με την απόσταση $d[s_S, s_A]$ στην ουρά προτεραιότητας που χρησιμοποιείται από τον αλγόριθμο του Dijkstra, αποθηκεύεται και η συσχετιζόμενη σύνδεση c' έτσι ώστε η ακμή $(d_{c'}, s_A)$ να είναι η εκείνη για την οποία προέκυψε η συγκεκριμένη απόσταση κατά το βήμα χαλάρωσης. Για τις υπόλοιπες κορυφές (που δεν είναι μεταγωγής) η συσχετιζόμενη σύνδεση θεωρείται ότι έχει την τιμή NIL.

Τώρα, δεδομένου ενός ερωτήματος NA (S, T, t_S) , ο αλγόριθμός DTM-Q λειτουργεί ως εξής.

Αρχικά, ξεκινά με την εισαγωγή των κορυφών μεταγωγής s_S του σταθμού αναχώρησης S στην ουρά προτεραιότητας PQ . Η απόσταση και η συσχετιζόμενη με την κορυφή s_S σύνδεση στην PQ τίθενται στην τιμή 0 και NIL αντίστοιχα. Κατόπιν, ενώ η PQ δεν είναι άδεια, ο αλγόριθμος διαγράφει κορυφές μια προς μια από την PQ , όπως ο κλασικός αλγόριθμος Dijkstra. Όταν μια κορυφή διαγράφεται από την PQ , η συμπεριφορά του αλγορίθμου εξαρτάται από το είδος της διαγραφόμενης κορυφής.

Η επεξεργασία των κορυφών αναχώρησης είναι παρόμοια με εκείνη του κλασικού αλγορίθμου Dijkstra, δηλαδή:

- Χαλαρώνονται οι εξερχόμενες ακμές της κορυφής αναχώρησης με σκοπό να βρεθεί μια καλύτερη διαδρομή (αν υπάρχει).
- Οι συσχετιζόμενες (μέσω των εξερχόμενων ακμών) γειτονικές κορυφές αναχώρησης ή μεταγωγής, είτε εισάγονται στην ουρά προτεραιότητας (αν δεν είναι ήδη εκεί), είτε μειώνεται η τιμή της απόστασής τους (εφόσον αυτό είναι δυνατόν).

Η επεξεργασία των *κορυφών μεταγωγής* είναι πιο πολύπλοκη και έχει ως εξής.

Θεωρήστε τη χρονική στιγμή που η κορυφή μεταγωγής s_A ενός σταθμού $A \in \mathcal{B}$ διαγράφεται από την ουρά προτεραιότητας PQ . Έστω $d[s_S, s_A]$ η απόσταση από την κορυφή s_S στην s_A που συσχετίζεται με την s_A στην PQ και έστω c' η συσχετιζόμενη στοιχειώδης σύνδεση η οποία παράγει την $d[s_S, s_A]$. Η τιμή $d[s_S, s_A]$ ουσιαστικά αντιπροσωπεύει τον ελάχιστο χρόνο που απαιτείται για να φτάσει ένας επιβάτης στον σταθμό A από τον σταθμό S , αναχωρώντας την χρονική στιγμή t_S . Επομένως, ο αλγόριθμος πρώτα υπολογίζει την τιμή $x = t_d(c') + wt(d_{c'}, s_A) \pmod{1440}$, η οποία αναπαριστά τον χρόνο άφιξης της σύνδεσης c' . Αν η c' είναι ίση με NIL , τότε ο αλγόριθμος αναθέτει $x = t_S$.

Στη συνέχεια, για κάθε εξερχόμενη ακμή μεταγωγής (s_A, d_c) για την οποία $S_d(c) = A$ και $t_d(c) \geq x$, ο αλγόριθμος εκτελεί την λεγόμενη φάση *ενεργοποίησης*.

Μια ακμή (s_A, d_c) *ενεργοποιείται* αν $D_S[S_a(c)] = \text{false}$ και

$$\Delta(x, t_d(c)) = t_d(c) - x \pmod{1440} \geq \begin{cases} 0 & \text{αν } Z(c) = Z(c') \\ \text{transfer}(A) & \text{αλλιώς} \end{cases} \quad (4.1)$$

Η διαδικασία *ενεργοποίησης* μιας ακμής (s_A, d_c) αποτελείται από τα εξής βήματα:

- Θέτει $wt(s_A, d_c) = \Delta(x, t_d(c))$.
- Είτε εισάγει την κορυφή d_c στην ουρά προτεραιότητας (αν δεν ανήκει ήδη εκεί), είτε μειώνει την απόστασή της (εφόσον αυτό είναι δυνατόν).

Η φάση της ενεργοποίησης σταματά (επομένως και η εκτέλεση του αλγορίθμου) όταν το διάνυσμα D_A περιέχει μόνο τιμές αλήθειας, οι οποίες τίθενται στην ακόλουθη περίπτωση.

Έστω (s_A, d_c) η ακμή μεταγωγής με το μικρότερο χρόνο άφιξης η οποία ενεργοποιήθηκε για κάποιο σταθμό $S' = S_a(c)$. Τότε, ο αλγόριθμος θέτει $D_A[S'] = \text{true}$, όταν εξετάζεται μια ακμή $(s_A, d_{c'})$ για την οποία $S_a(c') = S'$ και $\Delta(t_a(c), t_a(c'))(\bmod 1440) > \text{transfer}(S')$ (τα χρονικά στιγμιότυπα $t_a(c)$ και $t_a(c')$ που απαιτούνται για τον έλεγχο μιας τέτοιας συνθήκης μπορούν να υπολογιστούν χρησιμοποιώντας τις τιμές x , $t_d(c)$, $t_d(c')$ και τα συσχετιζόμενα κόστη ακμών). Το γεγονός ότι οι κορυφές αναχώρησης διατάσσονται με βάση το χρόνο άφιξης διασφαλίζει ότι ο αλγόριθμος μπορεί να αγνοήσει οποιαδήποτε σύνδεση που αναχωρεί από τον A και κατευθύνεται προς τον S' μετά από αυτό το βήμα, αφού κάτι τέτοιο δεν μπορεί να οδηγήσει σε καλύτερη λύση για το συγκεκριμένο ερώτημα ΝΑ.

Με άλλα λόγια, αν δύο ακμές μεταγωγής (s_A, d_{c_1}) και (s_A, d_{c_2}) , που αντιστοιχούν σε δύο στοιχειώδεις συνδέσεις c_1 και c_2 , οδηγούν στον ίδιο σταθμό B , ικανοποιούν την ανισότητα (4.1), και έχουν δύο τιμές άφιξης που διαφέρουν κατά μία τιμή μεγαλύτερη από $\text{transfer}(B)$, τότε επιλέγεται (δηλαδή ενεργοποιείται) μόνο εκείνη με το μικρότερο χρόνο άφιξης ενώ η άλλη απορρίπτεται, αφού δεν μπορεί να οδηγήσει σε καλύτερη λύση άφιξης στον B . Αυτό επιτυγχάνεται με τον κατάλληλο ορισμό των κοστών και των τιμών του D_A . Πράγματι, αν υποθέσουμε ότι $x \leq \min\{t_d(c_1), t_d(c_2)\}$ και $t_a(c_1) < t_a(c_2) + \text{transfer}(B)(\bmod 1440)$, δηλαδή, ότι $t_a(c_2)$ είναι ο μικρότερος χρόνος άφιξης που ικανοποιεί την παραπάνω συνθήκη, τότε $D_A[B] = \text{true}$ όταν εξετάζεται η ακμή (s_A, d_{c_2}) και στα κόστη εκχωρούνται οι τιμές $wt(s_A, d_{c_1}) = t_d(c_1) - x$ και $wt(s_A, d_{c_2}) = \infty$. Ως εκ τούτου, η d_{c_2} δεν θα είναι ποτέ μέρος μιας λύσης του συγκεκριμένου ερωτήματος ΝΑ. Σημειώνεται ότι αν προκύψουν δύο διαδρομές με το ίδιο ελάχιστο κόστος, επιλέγεται αυθαίρετα μία από τις δύο.

Η εκτέλεση του αλγορίθμου DTM-Q τερματίζεται μόλις η κορυφή μεταγωγής s_T , που συσχετίζεται με τον σταθμό άφιξης T , διαγράφεται από την ουρά προτεραιότητας PQ . Ο αλγόριθμος επιστρέφει ως χρόνο άφιξης t_T στον T το άθροισμα $t_T = d[s_S, s_T] + t_S$.

Παράδειγμα. Για να γίνει πιο κατανοητή η συμπεριφορά του αλγορίθμου DTM-Q, παρατίθεται ένα παράδειγμα εκτέλεσης του αλγόριθμου στο γράφημα του Σχήματος 4.3 για το ερώτημα NA $(S, T, t_S) = (B, C, 13)$. Η υπόθεση που γίνεται είναι ότι $transfer(A) = transfer(B) = transfer(C) = 5$ λεπτά.

Ο αλγόριθμος ξεκινά με την εισαγωγή στην PQ της s_B με απόσταση 0 και με την προεπιλεγμένη τιμή συσχετιζόμενης σύνδεσης. Επιπλέον, θέτει $transfer(B) = 0$. Στη συνέχεια, ενόσω η PQ δεν είναι άδεια, συνεχίζει τη διαγραφή κορυφών από την PQ μία προς μία. Όταν η s_B διαγράφεται με απόσταση $d[s_B] = 0$, οι εξερχόμενες ακμές της εξετάζονται, ξεκινώντας από αυτήν που οδηγεί σε μια κορυφή αναχώρησης d_i της οποίας ο χρόνος αναχώρησης t_{d_i} είναι μεγαλύτερος ή ίσος με $x = t_S = 13$. Στο παράδειγμά μας, η πρώτη κορυφή αυτού του είδους είναι η μοναδική με την ετικέτα 15 στο Σχήμα 4.3. Έστω ότι αυτή η κορυφή ονομάζεται b_1 , με $t_{b_1} = 15$, και έστω ότι η συσχετιζόμενη σύνδεση είναι η c_1 (ακμή $(15, s_C)$). Σε αυτό το σημείο ξεκινά η φάση της ενεργοποίησης. Συγκεκριμένα, αρχικά ενεργοποιείται η ακμή μεταγωγής που οδηγεί στην b_1 θέτοντας το κόστος της ισό με $\Delta(x, t_{b_1}) = 15 - 13 = 2 \geq transfer(B)$ και εισάγοντας την b_1 στην PQ με απόσταση 2.

Κατόπιν, εξετάζεται η επόμενη ακμή, συγκεκριμένα εκείνη που οδηγεί στην κορυφή με ετικέτα 25 στο Σχήμα 4.3. Έστω ότι αυτή η κορυφή ονομάζεται b_2 , με $t_{b_2} = 25$, και έστω ότι η συσχετιζόμενη σύνδεση της είναι η c_2 (ακμή $(25, s_C)$). Παρατηρήστε ότι αυτή η σύνδεση ενώνει μια κορυφή αναχώρησης του σταθμού B και πάλι με την κορυφή μεταγωγής του σταθμού C . Η ακμή μεταγωγής (s_B, b_2) έχει κόστος $\Delta(x, t_{b_2}) = 25 - 13 = 12 \geq transfer(B)$. Επομένως, σύμφωνα με την ανισότητα (4.1), ελέγχεται αν η ακμή (s_B, b_2) είναι τέτοια ώστε $S_a(c_2) = C$ και $\Delta(t_a(c_1), t_a(c_2)) \pmod{1440} > transfer(C)$. Παρατηρήστε ότι $t_a(c_1) = t_{b_1} + wt(c_1) = 15 + 5 = 20$, $t_a(c_2) = t_{b_2} + wt(c_2) = 25 + 8 = 33$, και $\Delta(t_a(c_1), t_a(c_2)) \pmod{1440} = 13$, το οποίο είναι ξεκάθαρα μεγαλύτερο από το $transfer(C) = 5$. Επομένως, έχει όντως βρεθεί μία σύνδεση που οδηγεί στον ίδιο σταθμό C , ικανοποιεί την ανισότητα (4.1), και έχει δύο χρόνους άφιξης στον C που η διαφορά τους είναι μεγαλύτερη από το $transfer(C)$. Άρα, τίθεται $D_B[C] = \text{true}$ και τερματίζεται η αναζήτηση για όλες τις υπόλοιπες συνδέσεις που οδηγούν στον σταθμό C , αφού καμία από αυτές δεν μπορεί να δώσει καλύτερη λύση άφιξης στον C , δηλαδή δεν εξετάζεται καμία περαιτέρω ακμή

σύνδεσης (s_B, b_i) και η αντίστοιχη σύνδεσή της c_i για την οποία $S_a(c_i) = C$.

Η εκτέλεση του αλγορίθμου συνεχίζεται με τη διαγραφή της b_1 από την PQ και τη χαλάρωση των εξερχόμενων ακμών της, η οποία συνεπάγεται την εισαγωγή στην PQ αφενός της κορυφής s_C με απόσταση $d[s_B, b_1] + wt(c_1) = 2 + 5 = 7$, και αφετέρου της κορυφής αναχώρησης με ετικέτα 26 στο Σχήμα 4.3, εξαιτίας της χαλάρωσης της εξερχόμενης ακμής οχήματος, έστω e_1 (ακμή (15,26) στο Σχήμα 4.3), της b_1 με απόσταση $d[s_B, b_1] + wt(e_1) = 2 + 11 = 13$.

Το επόμενο βήμα του αλγόριθμου αποτελείται από τη διαγραφή της s_C , η οποία ενεργοποιεί το κριτήριο τερματισμού του αλγορίθμου DTM-Q, αφού η κορυφή μεταγωγής s_C που συσχετίζεται με τον επιθυμητό σταθμό άφιξης C διαγράφεται από την PQ . Έτσι, ο αλγόριθμος επιστρέφει μια διαδρομή συνολικού (ελάχιστου) μήκους 7, η οποία συνεπάγεται τον αντίστοιχο χρόνο άφιξης $t_C = 13 + 7 = 20$ στον σταθμό C . ■

Ο αλγόριθμος DTM-Q μπορεί να τροποποιηθεί κατάλληλα προκειμένου να απαντά σε ερωτήματα EAM (S, T, t_S) . Η τροποποίηση έχει ως εξής. Πρώτον, δεν χρειάζεται η διατήρηση και αποθήκευση του διανύσματος D . Δεύτερον, όταν μία κορυφή μεταγωγής s_A με συσχετιζόμενη σύνδεση c' διαγράφεται από την ουρά προτεραιότητας, τότε ενεργοποιούνται όλες οι εξερχόμενες ακμές μεταγωγής που ικανοποιούν τον περιορισμό του χρόνου μετεπιβίβασης (ανισότητα 4.1) και το κόστος κάθε τέτοιας ακμής μεταγωγής (s_A, d_c) τίθεται σε 0, αν $Z(c) = Z(c')$, ή σε 1, διαφορετικά.

Η χρονική πολυπλοκότητα του αλγορίθμου DTM-Q παρουσιάζεται από το παρακάτω θεώρημα.

Θεώρημα 8 *Ο αλγόριθμος DTM-Q επιλύει τα προβλήματα NA και EAM σε χρόνο χειρότερης περίπτωσης $O(|C| \log |C|)$ στο γραφοκεντρικό μοντέλο DTM.*

Απόδειξη. Δεδομένου ενός ερωτήματος NA (S, T, t_S) , η διαδρομή που προτείνεται από τον αλγόριθμο DTM-Q είναι, από την κατασκευή του γραφήματος DTM, ένα δρομολόγιο που ξεκινά από τον σταθμό S την χρονική στιγμή $t \geq t_S$ και φτάνει στον T την χρονική στιγμή t_T .

Μένει να αποδειχθεί ότι το δρομολόγιο αυτό έχει το ελάχιστο μήκος. Η απόδειξη αυτού του ισχυρισμού μιμείται την απόδειξη ορθότητας του αλγορίθμου

Dijkstra. Πιο συγκεκριμένα, αποδεικνύεται ότι όταν μια κορυφή μεταγωγής s_A διαγράφεται από την ουρά προτεραιότητας PQ του αλγορίθμου Dijkstra, η τιμή $d[s_S, s_A]$ είναι η ελάχιστη, παράγοντας επομένως ένα δρομολόγιο ελάχιστης διάρκειας από τον S στο A .

Θα χρησιμοποιηθεί η μέθοδος της απαγωγής σε άτοπο. Έστω ότι ο A είναι ένας σταθμός τέτοιος ώστε η κορυφή s_A είναι η πρώτη κορυφή που διαγράφεται από την ουρά προτεραιότητας PQ και η ετικέτα $d[s_S, s_A]$ της οποίας δεν είναι η ελάχιστη. Προφανώς, $S_A \neq S_S$. Όταν η κορυφή s_A εισέρχεται για πρώτη φορά στην PQ , ή μειώνεται η τιμή της για τελευταία φορά, αυτό συμβαίνει εξαιτίας της χαλάρωσης μιας ακμής σύνδεσης $(d(c'), s_A)$ και μιας ακμής μεταγωγής $(s_B, t_d(c'))$ για κάποιον σταθμό B και σύνδεση c' . Κατά την διάρκεια της φάσης ενεργοποίησης, η ακμή μεταγωγής $(s_B, t_d(c'))$ ενεργοποιείται (δηλαδή το κόστος της τίθεται σε μια πεπερασμένη τιμή) μόνο αν, ανάμεσα σε όλες τις πιθανές συνδέσεις που συνδέουν τον σταθμό B με τον σταθμό A και οι οποίες ικανοποιούν την ανισότητα (4.1), η c' είναι η μοναδική που έχει τη ελάχιστη διάρκεια $\Delta(c')$. Επιπλέον, αφού η ετικέτα (προσωρινή απόσταση) της s_A έχει ενημερωθεί για τελευταία φορά εξαιτίας της χαλάρωσης της ακμής $(d(c'), s_A)$, τότε όλες οι υπόλοιπες ακμές σύνδεσης που οδηγούν στον s_A αντιπροσωπεύουν μεγαλύτερες σε διάρκεια διαδρομές. Επομένως, πρέπει να ισχύει ότι όταν η κορυφή μεταγωγής s_B του σταθμού B διαγράφεται από την PQ , η απόσταση $d[s_S, s_B]$ δεν είναι η ελάχιστη, το οποίο αντιφάσκει με το γεγονός ότι η s_A ήταν η πρώτη κορυφή που είχε διαγραφεί με μη-ελάχιστη απόσταση.

Η χρονική πολυπλοκότητα του αλγορίθμου DTM-Q είναι ίδια με αυτήν του αλγορίθμου του Dijkstra σε ένα γράφημα με $n = |\mathcal{B}| + |\mathcal{C}|$ κορυφές και $m \leq 3|\mathcal{C}|$ ακμές, συν το επιπλέον χρονικό κόστος για το βήμα της ενεργοποίησης. Το βήμα αυτό, στη χειρότερη περίπτωση, πρέπει να εκτελεστεί για κάθε εξερχόμενη ακμή μιας κορυφής μεταγωγής s_A όταν διαγράφεται από την ουρά προτεραιότητας και απαιτεί τη διερεύνηση ενός στοιχείου στο διάνυσμα D_A . Επομένως, η χαλάρωση μιας ακμής απαιτεί χρόνο $O(\log |D_A|)$ στον αλγόριθμο DTM-Q. Αφού το D_A έχει το πολύ $|\mathcal{B}|$ στοιχεία, η χαλάρωση απαιτεί το πολύ $O(\log |\mathcal{B}|)$ χρόνο για κάθε ακμή στο γράφημα DTM. Συνολικά λοιπόν, ο χρόνος όλων των χαλαρώσεων ακμών είναι (στη χειρότερη περίπτωση) $O(m \log |\mathcal{B}|)$, και επομένως η χρονική πολυπλοκότητα του DTM-Q είναι

$O(n \log n + m \log |\mathcal{B}|) = O(|\mathcal{C}| \log |\mathcal{C}|)$, αφού $|\mathcal{B}| \leq |\mathcal{C}|$.

Η ορθότητα και η πολυπλοκότητα του τροποποιημένου αλγόριθμου DTM-Q για απάντηση ερωτημάτων EAM προκύπτει άμεσα από την απόδειξη που μόλις παρουσιάστηκε για το πρόβλημα NA. ■

4.2.3 Αλγόριθμος Ενημέρωσης Καθυστερήσεων

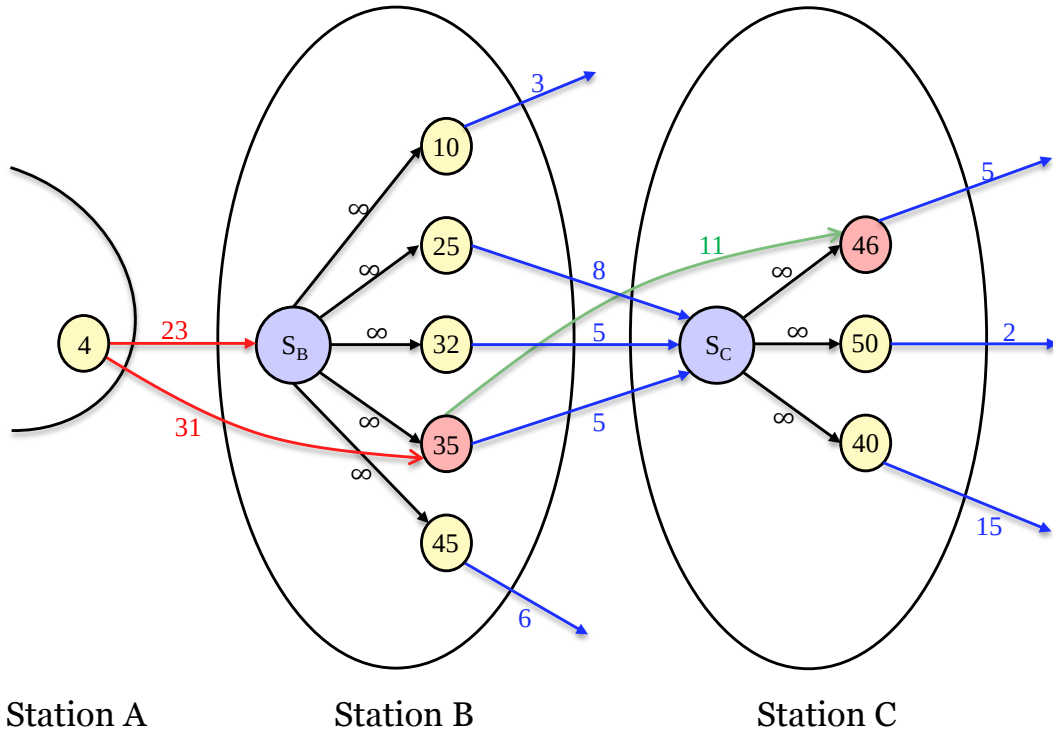
Στην ενότητα αυτή παρουσιάζεται ο αλγόριθμος ενημέρωσης DTM-U για τη διαχείριση καθυστερήσεων που μπορούν να συμβούν σε ένα πίνακα δρομολογίων και για την ενσωμάτωση των καθυστερήσεων αυτών στο αντίστοιχο γράφημα DTM.

Έστω ένας δεδομένος πίνακας δρομολογίων $\mathcal{T}$, μια καθυστέρηση γ λεπτών σε μια σύνδεση c του $\mathcal{T}$, και έστω $\mathcal{T}'$ ο αντίστοιχος *αναθεωρημένος πίνακας δρομολογίων* (disposition timetable).

Αν ο $\mathcal{T}$ αναπαριστάνεται από ένα γράφημα DTM G , τότε ο αλγόριθμος ενημέρωσης καθυστερήσεων DTM-U υπολογίζει ένα γράφημα DTM G' , το οποίο αναπαριστάνει τον $\mathcal{T}'$.

Για λόγους σαφήνειας και προκειμένου να καταδειχτεί καλύτερα πώς λειτουργεί ο DTM-U, θα παρουσιαστεί ο αλγόριθμος μαζί με ένα παράδειγμα (σε αντιστοιχία με τον αλγόριθμο ενημέρωσης TE-UH της Ενότητας 4.1.3). Το παράδειγμα αφορά τον σταθμό B του Σχήματος 4.3 (πριν συμβεί μια καθυστέρηση) και του Σχήματος 4.4 (αφού συμβεί η καθυστέρηση), υποθέτοντας ότι ένα όχημα φτάνει στον σταθμό B με καθυστέρηση $\gamma = 20$ λεπτών.

Πρώτον, το κόστος της ακμής σύνδεσης $(d_c, s_{S'})$ και της ακμής οχήματος $(d_c, d_{\hat{c}})$ ενημερώνεται με την προσθήκη της ποσότητας $\gamma \pmod{1440}$ στην τιμή τους, όπου $S' = S_a(c)$ και $d_{\hat{c}}$ είναι η κορυφή αναχώρησης του S' έτσι ώστε $Z(c) = Z(\hat{c})$ (αν υπάρχει). Ένα παράδειγμα αυτής της ενημέρωσης φαίνεται στο Σχήμα 4.4, όπου το κόστος της ακμής σύνδεσης που συνδέει τις κορυφές αναχώρησης 4 του σταθμού A με την κορυφή μεταγωγής s_B του σταθμού B αυξάνεται κατά $\gamma = 20$ και γίνεται 23, ενώ το κόστος της ακμής οχήματος που συνδέει την ίδια κορυφή αναχώρησης του σταθμού A με την κορυφή αναχώρησης 15 στο Σχήμα 4.3 (35 στο Σχήμα 4.4) του σταθμού B , αυξάνεται κατά $\gamma = 20$ και γίνεται 31.



Σχήμα 4.4: Διαχείριση καθυστερήσεων στο μοντέλο DTM. Η καθυστέρηση $\gamma = 20$ λεπτών προκαλεί αλλαγές στα κόστη δύο ακμών (κόκκινων) και την ενημέρωση του συσχετιζόμενου χρόνου με τις αντίστοιχες κορυφές αναχώρησης (κόκκινες).

Δεύτερον, για κάθε σύνδεση c' η οποία επηρεάζεται από την καθυστέρηση που προκλήθηκε στην c , έτσι ώστε ο χρόνος αναχώρησης στον $\mathcal{T}$ να διαφέρει από τον αντίστοιχο στον $\mathcal{T}'$, τότε ο συσχετιζόμενος χρόνος της κορυφής αναχώρησης $d_{c'}$ ενημερώνεται αυξάνοντάς τον κατά $\gamma \pmod{1440}$. Ένα παράδειγμα αυτής της ενημέρωσης φαίνεται στο Σχήμα 4.4 όπου οι κορυφές αναχώρησης των σταθμών B και C , που αρχικά είχαν χρόνους αναχώρησης 15 και 26 αντίστοιχα (δείτε Σχήμα 4.3), ενημερώνονται στους (καθυστερημένους) χρόνους αναχώρησης 35 και 46 από τους σταθμούς B και C , αντίστοιχα.

Να σημειωθεί ότι η αναζήτηση των επηρεασμένων (απο την καθυστέρηση) συνδέσεων c' μπορεί να γίνει εκτελώντας μια διερεύνηση του γραφήματος DTM G , ξεκινώντας από την κορυφή αναχώρησης d_c και επιλέγοντας κατά τη διάρκεια της διερεύνησης όλες τις συνδέσεις c' για τις οποίες $Z(c') = Z(c)$.

Επιπλέον, απαιτείται ένας περαιτέρω υπολογισμός στην περίπτωση που το G χρησιμοποιείται για απάντηση ερωτημάτων ΝΑ. Σε αυτή την περίπτωση, η ενημέρωση των κοστών των ακμών και των χρόνων είναι πιθανό να χαλάσει τη διάταξη μέσα στον πίνακα που αναπαριστάνει τις ακμές. Για να αποκατασταθεί η διάταξη, αρκεί να συγκρίνουμε τις νέες τιμές των χρόνων άφιξης των λίγων ενημερωμένων ακμών και να τις αντιμετωπίσουμε εντός του πίνακα, αν χρειάζεται. Για παράδειγμα, η κορυφή αναχώρησης 15 του σταθμού B στο Σχήμα 4.3 μετακινείται δύο θέσεις προς τα κάτω στο Σχήμα 4.4 προκειμένου να αποκατασταθεί η σωστή διάταξη σε σχέση με τον χρόνο άφιξης στο σταθμό C , ενώ η κορυφή αναχώρησης 26 του σταθμού C στο Σχήμα 4.3 διατηρεί τη θέση της (46 στο Σχήμα 4.4) παρά την αύξηση.

Παρατηρήστε ότι η εκτέλεση του αλγόριθμου $DTM-U$ δεν επιφέρει καμία τοπολογική αλλαγή στο γράφημα G . Η χρονική πολυπλοκότητα του αλγορίθμου παρουσιάζεται από το παρακάτω θεώρημα.

Θεώρημα 9 *Ο αλγόριθμος $DTM-U$ απαιτεί χρόνο χειρότερης περίπτωσης $O(|C|)$ για να ενημερώσει έναν πίνακα δρομολογίων που αναπαριστάνεται από το γραφοκεντρικό μοντέλο DTM μετά από μια καθυστέρηση οχήματος.*

Απόδειξη. Έστω T ο αρχικός πίνακας δρομολογίων που αναπαριστάνεται από ένα γράφημα DTM G και έστω T' ο αντίστοιχος αναθεωρημένος πίνακας δρομολογίων μετά από μια καθυστέρηση οχήματος κατά γ λεπτά.

Το πρώτο μέρος της απόδειξης στοχεύει να δείξει ότι το γράφημα DTM G' , που προκύπτει μετά από την εκτέλεση του αλγορίθμου $DTM-U$ στο G , είναι το γράφημα DTM που αναπαριστά τον πίνακα δρομολογίων T' .

Υπενθυμίζεται ότι η εκτέλεση του αλγόριθμου δεν επιφέρει καμία τοπολογική αλλαγή στο γράφημα G . Επομένως, η ορθότητα της μεθόδου προκύπτει από τη σωστή ενημέρωση των συσχετιζόμενων χρονοσφραγίδων και κοστών των επηρεασμένων, από την καθυστέρηση γ , κορυφών και ακμών του G . Η ζητούμενη σωστή ενημέρωση προκύπτει από τις εξής παρατηρήσεις:

- (α) Όλες οι ακμές σύνδεσης και οχήματος του G' , οι οποίες συσχετίζονται με επηρεασμένες συνδέσεις c' , ενημερώνονται με τα σωστά κόστη σύμφωνα

με τον ορισμό του γραφήματος DTM. Πράγματι, για κάθε καθυστερημένη σύνδεση c , τα κόστη της ακμής σύνδεσης (d_c, s_A) αλλά και της ακμής οχήματος $(d_c, d_{\hat{c}})$ (αν υπάρχει) ενημερώνονται με την προσθήκη της ποσότητας $\gamma(\text{mod } 1440)$ στην τιμή τους.

- (β) Οι κορυφές αναχώρησης έχουν τις κατάλληλες συσχετιζόμενες χρονοσφραγίδες. Πράγματι, οι χρονοσφραγίδες των κορυφών αναχώρησης $d_{c'}$ των επηρεασμένων συνδέσεων c' αυξάνονται κατά την ίδια ποσότητα $\gamma(\text{mod } 1440)$.
- (γ) Στην περίπτωση που το γράφημα DTM χρησιμοποιείται για να λύσει το ΝΑ, οι κορυφές αναχώρησης διατάσσονται με βάση το νέο χρόνο άφιξης (στον αμέσως επόμενο σταθμό). Σε αυτή την περίπτωση, είναι εύκολο να αποκατασταθεί η διάταξη του πίνακα ακμών μετακινώντας τις ενημερωμένες ακμές στην κατάλληλη θέση, σύμφωνα με τη (νέα) ενημερωμένη ώρα άφιξης.

Το δεύτερο μέρος της απόδειξης στοχεύει να δείξει ότι ο αλγόριθμος DTM-U υπολογίζει το G' σε χρόνο χειρότερης περίπτωσης $O(|\mathcal{C}|)$.

Έστω m_i ο αριθμός των εξερχόμενων ακμών από την κορυφή μεταγωγής s_i του γραφήματος DTM G , για κάθε $i \in \mathcal{B}$. Αν ο G χρησιμοποιείται για την απάντηση ερωτημάτων ΕΑΜ, τότε η εύρεση των επηρεασμένων συνδέσεων c' και οι αντίστοιχες ενημερώσεις χρονοσφραγίδων και κοστών απαιτεί χρόνο $O(\sum_{i \in \mathcal{B}} m_i) = O(m) = O(|\mathcal{C}|)$ από τον αλγόριθμο DTM-U. Διαφορετικά, αν το G χρησιμοποιείται για την απάντηση ερωτημάτων ΝΑ, τότε πρέπει επιπλέον να συνυπολογιστεί και ο χρόνος επαναδιάταξης του πίνακα ακμών. Αυτός ο επιπλέον χρόνος είναι επίσης $O(\sum_{i \in \mathcal{B}} m_i) = O(m) = O(|\mathcal{C}|)$ χρησιμοποιώντας ταξινόμηση βάσης (radix sort) για την επαναδιάταξη, αφού τα κόστη ακμών είναι ακέραιοι. Παρατηρήστε όμως ότι επειδή μόνο ένα στοιχείο είναι εκτός διάταξης σε κάθε πίνακα, το ίδιο αποτέλεσμα μπορεί να επιτευχθεί μετακινώντας στη σωστή θέση του πίνακα το μοναδικό στοιχείο του οποίου η χρονοσφραγίδα έχει αλλάξει. Η διαδικασία αυτή απαιτεί χρόνο $O(m_i)$ για κάθε κορυφή μεταγωγής s_i , δηλαδή και πάλι συνολικό χρόνο $O(\sum_{i \in \mathcal{B}} m_i) = O(m) = O(|\mathcal{C}|)$. ■

Το Θεώρημα 9 δίνει ένα άνω όριο το οποίο απέχει πολύ από την πραγματικότητα, αφού οι σταθμοί που ενημερώνουν χρονοσφραγίδες και κόστη ακμών είναι πολύ λιγότερο από $|B|$, ειδικά χάρις στον ανθεκτικό σχεδιασμό των σύγχρονων πινάκων δρομολογίων [16, 17, 18, 39, 62, 74]. Το κεφάλαιο 5, το οποίο παρουσιάζει την εκτενή πειραματική αξιολόγηση των αλγορίθμων, καταδεικνύει ότι στην πράξη ο αριθμός των ακμών που πρέπει να ενημερωθεί σαν συνέπεια μιας καθυστέρησης είναι πάντα ένα μικρός σταθερός αριθμός σε σχέση με το μέγεθος του γραφήματος.

4.2.4 Επιτάχυνση Αλγορίθμου Απάντησης Ερωτημάτων

Στην ενότητα αυτή παρουσιάζεται μία τροποποιημένη έκδοση του βασικού αλγορίθμου DTM-Q απόκρισης ερωτημάτων, η οποία εκμεταλλεύεται την κατασκευή του μοντέλου και ορισμένους περιορισμούς για να μειώσει περαιτέρω το χρόνο απόκρισης. Η νέα έκδοση ονομάζεται DTM-QH και βασίζεται στην ενδυνάμωση του βασικού αλγορίθμου με μια σειρά ευρετικών τεχνικών.

Πρώτον, εφαρμόζεται στον DTM-Q μια τροποποίηση της γενικευμένης τεχνικής ομαδοποίησης κορυφών [27]. Σκοπός της τεχνικής αυτής είναι η περικοπή της αναζήτησης μειώνοντας το μέγεθος της ουράς προτεραιότητας PQ του Dijkstra. Αυτό επιτυγχάνεται με την εκμετάλλευση της δομής του γραφήματος αναπαράστασης του πίνακα δρομολογίων που αποτρέπει την εισαγωγή στην PQ συνόλων κορυφών που σαφώς δεν αποτελούν μέρος των συντομότερων διαδρομών. Συγκεκριμένα, αν κατά την εκτέλεση του αλγορίθμου ερωτημάτων κορυφές, που ανήκουν σε έναν σταθμό τον οποίο έχει ήδη επισκεφθεί ο αλγόριθμος (δηλαδή η κορυφή μεταγωγής του σταθμού έχει ήδη διαγραφεί), εξετάζονται (είτε για πρώτη φορά, είτε όχι), τότε απορρίπτονται αμέσως και δεν εισάγονται στην ουρά προτεραιότητας. Αυτό συνεπάγεται μικρότερο συνολικά μέγεθος της ουράς προτεραιότητας και άρα συνολικά μειωμένο αριθμό λειτουργιών που σχετίζονται με την PQ (π.χ. λειτουργίες εισαγωγής, μείωσης τιμής κλειδιού, διαγραφής στοιχείου με το ελάχιστο κλειδί). Η ορθότητα σε αυτή την περίπτωση μπορεί εύκολα να διασφαλιστεί. Πράγματι, αν η κορυφή μεταγωγής ενός σταθμού έχει διαγραφεί από την PQ , τότε υπάρχει βεβαιότητα ότι, λόγω του βέλτιστου των υποδιαδρομών μιας συντομότερης διαδρομής,

δεν μπορεί να βρεθεί καμία διαδρομή συντομότερη από αυτήν που έχει ήδη βρεθεί. Σημειώνεται ότι αυτή η προσέγγιση είναι κάπως παρόμοια με το τροποποιημένο κριτήριο εξερεύνησης κορυφών που εφαρμόστηκε στο ανηγμένο χρονοεκτεταμένο γράφημα (Ενότητα 4.1.2). Ωστόσο, στην περίπτωση του DTM-Q δεν χρειάζεται να αποθηκεύονται οι τρέχοντες χρόνοι άφιξης σε ένα σταθμό που έχει ήδη εξεταστεί, αφού οι κορυφές μεταγωγής αντιπροσωπεύουν όλα τα χρονικά γεγονότα άφιξης σε ένα σταθμό.

Δεύτερον, εφαρμόζεται στον DTM-Q μια παραλαγή της τεχνικής παραλείψης κορυφών για την μείωση του μεγέθους του χρονοεκτεταμένου γραφήματος [72] κατά το χρόνο εκτέλεσης του αλγορίθμου. Η αρχική τεχνική [72] στόχευε και επέτρεπε τη μόνιμη μείωση του μεγέθους του χρονοεκτεταμένου γραφήματος με την παράκαμψη (μέσω εισαγωγής δύο κατάλληλων ακμών) κορυφών με βαθμό εξόδου ίσο με 1. Τώρα, η τεχνική αυτή εφαρμόζεται στις κορυφές με βαθμό εξόδου 1 κατά το χρόνο εκτέλεσης του αλγορίθμου ως εξής: όταν η ρουτίνα απάντησης ερωτημάτων εξετάζει μία κορυφή x με βαθμό εξόδου 1, το βάθος χαλάρωσης επεκτείνεται σε δύο επίπεδα· δηλαδή, η κορυφή x δεν εισάγεται στην ουρά προτεραιότητας και εξετάζεται η κεφαλή (τελική κορυφή) της μοναδικής εξερχόμενης ακμής της x προκειμένου να διαπιστωθεί εάν βελτιώνει την τρέχουσα συντομότερη διαδρομή. Εύκολα διαπιστώνεται ότι αυτή η στρατηγική δεν επηρεάζει την ορθότητα του αλγορίθμου ερωτημάτων, ενώ παρέχει μια σαφή μείωση του μέσου μεγέθους της ουράς προτεραιότητας και του αριθμού των συσχετιζόμενων λειτουργιών της (π.χ. λειτουργίες εισαγωγής, μείωσης τιμής κλειδιού, διαγραφής στοιχείου με το ελάχιστο κλειδί).

Τρίτον, εφαρμόζεται μια ελαφρώς διαφορετική τοπολογική προσέγγιση στον DTM-Q προκειμένου να μειωθεί η περιττή εξερεύνηση και εξέταση κορυφών και ακμών κατά τη διάρκεια εκτέλεσης του αλγορίθμου. Σε αυτήν την παραλλαγή, σε κάθε σταθμό οι ακμές μεταγωγής ομαδοποιούνται με βάση τον γειτονικό σταθμό άφιξης και στη συνέχεια, για κάθε τέτοια ομάδα, οι ακμές ταξινομούνται κατά ώρα άφιξης στον γειτονικό σταθμό. Επομένως, όταν ο αλγόριθμος επισκεφθεί έναν γειτονικό σταθμό άφιξης S , ολόκληρη η ομάδα των ακμών μεταγωγής που κατευθύνονται στον S μπορεί άμεσα να παραληφθεί. Επίσης κατά τον αλγόριθμο διαχείρισης καθυστερήσεων, η προσέγγιση αυτή μειώνει τις λειτουργίες αντιμετάθεσης ακμών οι οποίες τώρα περιορίζονται μόνο μέσα

στην ομάδα ακμών του επηρεαζόμενου σταθμού.

Σημειώνεται ότι όλες οι παραπάνω ευρετικές τεχνικές είναι ανεξάρτητες και συμβατές με τη χρήση άλλων τεχνικών επιτάχυνσης, όπως η ALT.

4.3 Σύγκριση των Χρονοεκτεταμένων Μοντέλων

Στην ενότητα αυτή πραγματοποιείται σύγκριση του αυθεντικού ρεαλιστικού χρονοεκτεταμένου μοντέλου με τα δύο νέα που αναπτύχθηκαν στα πλαίσια της παρούσας διδακτορικής διατριβής, δηλαδή του ανηγμένου χρονοεκτεταμένου μοντέλου και του μοντέλου DTM.

Στην περίπτωση των καθυστερήσεων, ο αλγόριθμος DTM-U απαιτεί τον ίδιο υπολογιστικό χρόνο χειρότερης περίπτωσης με αυτόν που χρειάζονται οι αλγόριθμοι ενημέρωσης και στα δύο χρονοεκτεταμένα μοντέλα. Όμως, τα χρονοεκτεταμένα μοντέλα απαιτούν, μετά την επαναδιάταξη των κορυφών άφιξης, αναχώρησης και μετεπιβίβασης, και τοπολογική μεταβολή (εισαγωγή/διαγραφή) ακμών του γραφήματος. Αυτή η ενέργεια μπορεί να συνεπάγεται μεγάλο υπολογιστικό χρόνο που εξαρτάται από τον τρόπο αποθήκευσης του γραφήματος. Αντιθέτως, ο αλγόριθμος DTM-U μπορεί να ενημερώνει τη δομή δεδομένων του σε περίπτωση καθυστερήσεων σε γραμμικό χρόνο, χωρίς οποιαδήποτε αλλαγή στην τοπολογία γραφήματος. Πράγματι, μια καθυστέρηση στον πίνακα δρομολογίων προκαλεί λίγες αλλαγές στα κόστη των ακμών και στην ενημέρωση του χρόνου που σχετίζεται με τις αντίστοιχες κορυφές αναχώρησης. Σημειώνεται ότι αυτή η λειτουργία μπορεί να απαιτεί, σε ορισμένες περιπτώσεις, ένα επιπλέον βήμα αναδιάταξης των κορυφών αναχώρησης των σταθμών που εμπλέκονται στις αλλαγές σε σχέση με τους νέους χρόνους άφιξης στους γειτονικούς σταθμούς.

Αν και το μοντέλο DTM καθώς και τα δύο χρονοεκτεταμένα μοντέλα απαιτούν ασυμπτωτικά την ίδια πολυπλοκότητα χώρου, το γράφημα DTM έχει μικρότερο αριθμό κορυφών και ακμών. Το αυθεντικό ρεαλιστικό χρονοεκτεταμένο μοντέλο έχει $3|C|$ κορυφές και τουλάχιστον $4|C|$ ακμές (Θεώρημα 1), το ανηγμένο χρονοεκτεταμένο μοντέλο έχει $2|C|$ κορυφές και τουλάχιστον $3|C|$ ακμές (Θεώρημα 4), ενώ το μοντέλο DTM έχει $|B| + |C|$ κορυφές και το πολύ $3|C|$

ακμές (Θεώρημα 7). Υπενθυμίζοντας ότι $|C| \geq |B|$, προκύπτει ότι το μοντέλο DTM είναι χωρο-αποδοτικότερο.

Όσον αφορά τους αλγόριθμους απάντησης ερωτημάτων, η διαπίστωση είναι ότι, αν και οι τρεις αλγόριθμοι απαιτούν τον ίδιο υπολογιστικό χρόνο χειρότερης περίπτωσης, ο αλγόριθμος DTM-Q εκτελεί το επιπλέον βήμα της ενεργοποίησης ακμών και τον υπολογισμό των κοστών των ακμών μεταγωγής.

4.4 Συνδυασμός Γραφοκεντρικών Μοντέλων με την Ευρετική Τεχνική ALT

Ένα από τα πλεονεκτήματα των γραφοκεντρικών μοντέλων για συστήματα πινάκων δρομολογίων, σε σχέση με τα διανυσματοκεντρικά μοντέλα, είναι ότι τα πρώτα μπορούν να εκμεταλλευτούν τις ευρύτατα διαδεδομένες ευρετικές τεχνικές επιτάχυνσης που αναπτύχθηκαν τα τελευταία χρόνια για την βελτίωση του χρόνου υπολογισμού συντομότερων διαδρομών σε άλλες σημαντικές εφαρμογές, όπως είναι για παράδειγμα η εύρεση βέλτιστων διαδρομών σε οδικά δίκτυα. Πολλές από αυτές τις τεχνικές μπορούν να προσαρμοστούν και να χρησιμοποιηθούν σε συνδυασμό με τα γραφοκεντρικά μοντέλα σε συστήματα αναπαράστασης πληροφορίας πινάκων δρομολογίων με σκοπό να βελτιωθεί ο χρόνος απάντησης ερωτημάτων.

Σε αυτό το πλαίσιο και με σκοπό τη βελτίωση της απόδοσης των αλγορίθμων απάντησης ερωτημάτων που παρουσιάστηκαν σε αυτό το κεφάλαιο, έγινε συνδυασμός τους με την ευρετική τεχνική ALT [47], μια από τις απλούστερες και πλέον διαδεδομένες τέτοιες τεχνικές. Στην Ενότητα 2.4 γίνεται αναλυτική παρουσίαση της τεχνικής αυτής. Ο βασικός σκοπός της στοχοκατευθυνόμενης τεχνικής ALT είναι η γρηγορότερη προώθηση των συντομότερων διαδρομών προς τον προορισμό τους. Πολλές πειραματικές μελέτες έχουν δείξει ότι η ALT δίνει πολύ καλή επιτάχυνση, παρά την απλότητα της, και ότι το μέγεθος της επιτάχυνσης εξαρτάται γενικά από τη δομή του γραφήματος.

Η προσαρμογή της τεχνικής ALT στους αλγόριθμους απάντησης ερωτημάτων των χρονοεκτεταμένων μοντέλων είναι σχετικά εύκολη και πολλές παραλλαγές έχουν ήδη προταθεί· δείτε για παράδειγμα την εργασία [27]. Για αυτό το

λόγο, δεν γίνεται περαιτέρω ανάλυση στην παρούσα διατριβή της προσαρμογής της τεχνικής ALT στους αλγορίθμους απάντησης ερωτημάτων στα χρονοεκτεταμένα μοντέλα.

Το μοντέλο DTM είναι διαφορετικό και έτσι παρουσιάζεται παρακάτω η προσαρμογή της τεχνικής ALT στον αλγόριθμο απόκρισης ερωτημάτων.

Όπως ήδη αναφέρθηκε, η βασική ιδέα της στοχοκατευθυνόμενης τεχνικής ALT είναι να καθοδηγήσει την εκτέλεση του αλγορίθμου Dijkstra σε περιορισμένο χώρο αναζήτησης προωθώντας την αναζήτηση προς τον προορισμό t του δεδομένου ερωτήματος. Αυτό επιτυγχάνεται με την προσθήκη ενός *έγκυρου δυναμικού* στην προτεραιότητα κάθε κορυφής που ανήκει στην ουρά προτεραιότητας.

Όπως εξηγείται στην Ενότητα 2.4, τα έγκυρα δυναμικά υπολογίζονται με βάση ένα σύνολο κορυφών $L \subseteq V$ που ονομάζονται *ορόσημα* (landmarks). Δεδομένου του συνόλου L των οροσήμων, το έγκυρο δυναμικό $\pi_t(u)$ μιας κορυφής $u \in V$ (ως προς τον προορισμό t) υπολογίζεται από τον τύπο (δείτε Πόρισμα 1 στην Ενότητα 2.4)

$$\pi_t(u) = \max_{\ell \in L} \{ \max \{ d[u, \ell] - d[t, \ell]; d[\ell, t] - d[\ell, u] \} \}$$

Όπως επίσης εξηγείται στην Ενότητα 2.4, το $\pi_t(u)$ είναι το καλύτερο κάτω φράγμα της απόστασης $d[u, t]$ και αυτό αρκεί για να αποδειχθεί η ορθότητα του αλγορίθμου συντομότερης διαδρομής με χρήση της τεχνικής ALT.

Είναι εύκολο να διαπιστωθεί ότι όσο πιο αυστηρά είναι τα κάτω φράγματα, τόσο πιο περιορισμένος είναι ο χώρος αναζήτησης, δηλαδή ο αλγόριθμος απάντησης ερωτημάτων γίνεται ταχύτερος. Επομένως, η επιλογή καλών οροσήμων που παρέχουν αυστηρά κάτω φράγματα είναι θεμελιώδους σημασίας στη φάση της προεπεξεργασίας της τεχνικής ALT. Να σημειωθεί ότι έχουν προταθεί πολλές προσεγγίσεις για την επιλογή καλών οροσήμων [47].

Για τον υπολογισμό καλών οροσήμων στο μοντέλο (γράφημα) DTM εφαρμόζεται μια προσέγγιση παρόμοια με εκείνη που προτείνεται στην εργασία [27]. Συγκεκριμένα, επιλέγονται ως ορόσημα οι κορυφές μεταγωγής, κάθε μία από τις οποίες αντιπροσωπεύει την ομάδα κορυφών άφιξης ενός σταθμού. Επομένως το κάτω φράγμα της απόστασης $d[s_A, s_B]$ μεταξύ δύο κορυφών μεταγωγής

s_A και s_B , υποδηλώνει τον ελάχιστο χρόνο ταξιδιού μεταξύ όλων των συνδέσεων από τον σταθμό A στον σταθμό B . Τα κάτω φράγματα απόστασης μπορούν να υπολογιστούν κατά τη διάρκεια μιας φάσης προεπεξεργασίας, εκτελώντας ένα αλγόριθμο εύρεσης συντομότερων διαδρομών από κάθε κορυφή μεταγωγής προς όλες τις υπόλοιπες κορυφές σε ένα γράφημα $|B|$ κορυφών, και ακμών μεταξύ εκείνων των κορυφών (σταθμών) για τις οποίες υπάρχει τουλάχιστον μια στοιχειώδης σύνδεση. Το κόστος μιας ακμής μεταξύ δύο κορυφών (σταθμών) είναι η μικρότερη χρονική διάρκεια μιας στοιχειώδους σύνδεσης μεταξύ αυτών των κορυφών. Τα πιο αυστηρά κάτω φράγματα, με αυτή τη μέθοδο, μπορούν να ληφθούν αποθηκεύοντας όλες τις $O(|B|^2)$ αποστάσεις μεταξύ των ζευγών των κορυφών (σταθμών). Αυτή η μέθοδος είναι ιδιαίτερα αποδοτική όταν οι σταθμοί είναι σχετικά λίγοι.

Τέλος, συνδυάζεται η τεχνική ALT μαζί με τις ευρετικές μεθόδους που ήδη περιγράφηκαν στην Ενότητα 4.1.2 για τον αλγόριθμο απάντησης ερωτημάτων TE-QH του ανηγμένου χρονοεκτεταμένου γραφήματος, αλλά και στην Ενότητα 4.2.4 για τον αλγόριθμο DTM-QH του γραφήματος DTM.

Οι αλγόριθμοι που προκύπτουν από τους συνδυασμούς αυτούς ονομάζονται TE-QH-ALT και DTM-QH-ALT, αντίστοιχα. Ο συνδυασμός των ευρετικών τεχνικών μειώνει κατά πολύ το μέγεθος του χώρου αναζήτησης και οδηγεί σε πολύ πιο αποδοτικούς αλγορίθμους, όπως καταδεικνύει το Κεφάλαιο 5.

Κεφάλαιο 5

Πειραματική Αξιολόγηση Νέων Μοντέλων

Σε αυτό το κεφάλαιο παρουσιάζεται μια εκτενής πειραματική μελέτη για την αξιολόγηση της απόδοσης των αλγορίθμων των γραφο-κεντρικών μοντέλων που παρουσιάστηκαν στο Κεφάλαιο 4. Τα πειράματα εκτελέστηκαν σε υπολογιστικό σύστημα με επεξεργαστή Intel Quad-core i5-2500K 3.30GHz CPU και κύρια μνήμη 12 GB. Οι υλοποιήσεις έγιναν σε γλώσσα προγραμματισμού C++ (gcc μεταγλωτιστής 4.6.3 με επίπεδο βελτιστοποίησης O3).

Για τα γραφοκεντρικά μοντέλα του Κεφαλαίου 4, υλοποιήθηκαν:

- (α) Οι αλγόριθμοι για την κατασκευή του γραφήματος.
- (β) Οι βασικοί αλγόριθμοι απάντησης ερωτημάτων καθώς και οι αντίστοιχες εκλεπτυσμένες ευρετικές εκδοχές τους, τόσο για το ανηγμένο χρονοεκτεταμένο μοντέλο, όσο και για το μοντέλο DTM.
- (γ) Οι αλγόριθμοι ενημέρωσης της δομής αναπαράστασης των γραφοκεντρικών μοντέλων στην περίπτωση καθυστερήσεων.
- (δ) Οι αλγόριθμοι απάντησης ερωτημάτων ενισχυμένοι με την ευρετική τεχνική ALT.

Όλες οι υλοποιήσεις χρησιμοποιούνται σε συνδυασμό με τη δυναμική δομή γραφήματων της εργασίας [64], η οποία διαχειρίζεται με αποδοτικό τρόπο

δυναμικές αλλαγές (εισαγωγές/διαγραφές κορυφών και ακμών) σε γραφήματα.

5.1 Δεδομένα Εισόδου και Παράμετροι

Στα πειράματα που ακολουθούν χρησιμοποιούνται τα εξής δεδομένα εισόδου (δείτε Πίνακες 5.1 και 5.2):

ΠΔ	$\mathcal{T}$	$ \mathcal{B} $	$ \mathcal{C} $	TE-real	
				$ \mathcal{V} $	$ \mathcal{E} $
HaCon	bts	716	12 689	38 067	63 445
	ks	1 865	44 744	134 232	223 720
	efz	2 198	41 613	124 839	208 065
	bvb	2 874	292 542	877 626	1 462 710
	ger	6 493	428 982	1 286 946	2 144 910
	eur	7 786	596 129	1 788 387	2 912 210
GTFS	Βικτώρια	2 230	403 709	1 211 127	2 018 545
	Μαδρίτη	4 636	2 070 490	6 211 470	10 352 450
	Αθήνα	7 897	1 694 069	5 082 207	8 436 181
	Ρώμη	8 560	2 727 672	8 183 016	13 550 426
	Βερολίνο	12 838	4 322 549	12 967 647	21 612 745
	Λονδίνο	20 843	14 064 967	42 194 901	70 324 835
	Ελβετία	31 213	7 097 772	21 293 316	33 961 786
	Σουηδία	50 387	5 884 019	17 652 057	29 169 177

Πίνακας 5.1: Εξεταζόμενοι πίνακες δρομολογίων (ΠΔ) και μεγέθη αντίστοιχων γραφημάτων για το ρεαλιστικό χρονοεκτεταμένο μοντέλο (TE-real).

- (i) Τριών ειδών πίνακες δρομολογίων τρένων (με τις ονομασίες efz, ger, και eur) και τριών ειδών πίνακες δρομολογίων λεοφορείων (με τις ονομασίες bts, ks, και bvb) που εξάγονται από ένα μεγάλο σύνολο δεδομένων, που παρέχεται από την εταιρεία HaCon [49] για επιστημονική χρήση.
- (ii) Οκτώ διαφορετικά σύνολα δεδομένων πινάκων δρομολογίων στην μορφή General Transit Feed Specification (GTFS) [42], που περιέχουν διαφορετικά είδη μέσων μαζικής μεταφοράς. Πιο συγκεκριμένα, αυτά τα σύνολα δεδομένων αφορούν τις μητροπολιτικές πόλεις της Αθήνας, της Ρώμης, της Βικτώριας, της Μαδρίτης, του Βερολίνου και του Λονδίνου, καθώς και δεδομένα πινάκων δρομολογίων των εθνικών δικτύων των χωρών της

Σουηδίας και της Ελβετίας. Τα δεδομένα για την πόλη του Λονδίνου αποκτήθηκαν από τον σύνδεσμο [81], ενώ τα υπόλοιπα από τον σύνδεσμο [80].

Για κάθε πίνακα δρομολογίων $\mathcal{T}$, δημιουργείται: (α) το ρεαλιστικό χρονοεκτεταμένο γράφημα (β) το ανηγμένο χρονοεκτεταμένο γράφημα και (γ) το γράφημα DTM. Για την αποθήκευση των γραφημάτων, χρησιμοποιείται η δομή δεδομένων γραφημάτων όπως παρουσιάζεται στην εργασία [64]. Για το γράφημα DTM, οι εξερχόμενες ακμές μεταγωγής (s_s, d_c) κάθε κορυφής μεταγωγής s_s ταξινομούνται σύμφωνα με το χρόνο άφιξης $t_a(c)$ (στον γειτονικό σταθμό) της στοιχειώδους σύνδεσης c που συσχετίζεται με την κορυφή d_c κατά αύξουσα διάταξη. Ως ουρά προτεραιότητας, χρησιμοποιείται ο κλασικός δυαδικός σωρός.

ΠΔ	$\mathcal{T}$	$ \mathcal{B} $	$ \mathcal{C} $	TE-red		DTM	
				$ V $	$ E $	$ V $	$ E $
HaCon	bts	716	12 689	25 378	49 862	13 405	38 067
	ks	1 865	44 744	89 488	175 536	46 609	134 232
	efz	2 198	41 613	83 226	159 290	43 811	124 839
	bvb	2 874	292 542	585 084	1 154 792	295 416	877 626
	ger	6 493	428 982	857 964	1 668 171	435 475	1 286 946
	eur	7 786	596 129	1 192 258	2 316 081	603 915	1 719 952
GTFS	Βικτώρια	2 230	403 709	807 418	1 604 359	405 939	1 200 658
	Μαδρίτη	4 636	2 070 490	4 140 980	8 200 875	2 075 124	6 130 385
	Αθήνα	7 897	1 694 069	3 388 138	6 742 112	1 701 966	5 048 136
	Ρώμη	8 560	2 727 672	5 453 344	10 822 754	2 736 232	8 097 361
	Βερολίνο	12 838	4 322 549	8 645 098	17 024 138	4 335 387	12 701 695
	Λονδίνο	20 843	14 064 967	28 129 934	55 758 468	14 085 810	41 837 355
	Ελβετία	31 213	7 097 772	14 195 544	26 841 306	7 128 985	20 521 440
	Σουηδία	50 387	5 884 019	11 768 038	23 285 158	5 934 406	17 402 734

Πίνακας 5.2: Εξεταζόμενοι πίνακες δρομολογίων (ΠΔ) και μεγέθη των αντίστοιχων γραφημάτων για το ανηγμένο (ρεαλιστικό) χρονοεκτεταμένο μοντέλο (TE-red) και το μοντέλο DTM.

Οι πίνακες 5.1, 5.2, 5.3 και 5.4 παρέχουν λεπτομερείς πληροφορίες σχετικά με τους εισαγόμενους πίνακες δρομολογίων. Στους πίνακες 5.1 και 5.2, καταγράφεται, για κάθε πίνακα δρομολογίων, ο αριθμός των σταθμών $|\mathcal{B}|$ και ο αριθμός των στοιχειωδών συνδέσεων $|\mathcal{C}|$ μεταξύ των σταθμών, όπως επίσης και το πλήθος των κορυφών $|V|$ και των ακμών $|E|$ του αντίστοιχου γραφήματος,

για κάθε μοντέλο.

Στους πίνακες 5.3 και 5.4 καταγράφεται, για κάθε πίνακα δρομολογίων, ο μέσος χρόνος μετεπιβίβασης που απαιτείται για την αλλαγή οχημάτων, ο μέσος αριθμός γειτονικών στάσεων ή σταθμών, και το ποσοστό των υπάρχοντων μέσων μεταφοράς σε σχέση με το συνολικό αριθμό των στοιχειωδών συνδέσεων. Στους πίνακες δρομολογίων της Μαδρίτης, της Αθήνας και της Ρώμης, ο χρόνος μετεπιβίβασης δεν ήταν διαθέσιμος για ορίσθηκε εμπειρικά στα 3 λεπτά.

$\mathcal{T}$	efz	ger	eur	bts	ks	bvb	Βικτώρια	Μαδρίτη	Αθήνα
Χρ. Μετεπιβίβασης	6.5	6.5	6.5	6.5	6.5	6.5	6.5	3	3
Βαθμός Εξόδου	2.2	2.7	2.7	2.5	2.7	2.3	1.1	1.4	1.2
Λεωφορεία	100%								
Τοπικά Τρένα	100%								
Προαστιακός Σιδ.									
Μετρό									
Τραμ									

Πίνακας 5.3: Χαρακτηριστικά πινάκων δρομολογίων: μέσος χρόνος μετεπιβίβασης (mins), μέσος βαθμός γειτνίασης στάσεων/σταθμών και ποσοστό μέσων μεταφοράς (% του συνολικού).

$\mathcal{T}$	Ρώμη	Βερολίνο	Λονδίνο	Ελβετία	Σουηδία
Χρ. Μετεπιβίβασης	3	0.7	0.8	0.3	0.2
Βαθμός Εξόδου	1.3	2.7	1.2	2.4	2.3
Λεωφορεία	95.6%	76%	98%	80%	92.7%
Τοπικά Τρένα	4.4%	11%	2%	12%	1.0%
Προαστιακός Σιδ.		4%		7%	0.4%
Μετρό				1%	2.2%
Τραμ		9%			3.7%

Πίνακας 5.4: Χαρακτηριστικά πινάκων δρομολογίων: μέσος χρόνος μετεπιβίβασης (mins), μέσος βαθμός γειτνίασης στάσεων/σταθμών και ποσοστό μέσων μεταφοράς (% του συνολικού).

Οι πίνακες 5.1 και 5.2 επιβεβαιώνουν την ανάλυση που έγινε για το εαλιστικό χρονοεκτεταμένο μοντέλο (Θεώρημα 1), το ανηγμένο χρονοεκτεταμένο μοντέλο (Θεώρημα 4), και για το μοντέλο DTM (Θεώρημα 7), σε σχέση με το μέγεθος του γραφήματος. Για κάθε πίνακα δρομολογίων $\mathcal{T} = (\mathcal{Z}, \mathcal{B}, \mathcal{C})$, διαπιστώθηκε ότι ο αριθμός των κορυφών είναι ακριβώς $3|\mathcal{C}|$, $2|\mathcal{C}|$, και $|\mathcal{B}| + |\mathcal{C}|$, ενώ αριθμός των ακμών είναι πάντα μικρότερος από $5|\mathcal{C}|$, $4|\mathcal{C}|$, και $3|\mathcal{C}|$ για το

$\mathcal{T}$	bts	ks	efz	bvb	ger	eur	Βικτώρια
Χώρος (GB)	0.002	0.014	0.019	0.032	0.161	0.177	0.019
Χρόνος (mins)	0.002	0.011	0.019	0.026	0.162	0.191	0.022

$\mathcal{T}$	Μαδρίτη	Αθήνα	Ρώμη	Βερολίνο	Λονδίνο	Ελβετία	Σουηδία
Χώρος (GB)	0.082	0.183	0.253	0.629	1.4	2.6	9.4
Χρόνος (mins)	0.063	0.153	0.249	0.729	1.785	4.004	15.648

Πίνακας 5.5: Απαιτούμενος χρόνος και χώρος προεπεξεργασίας από την ευρετική τεχνική ALT για όλους τους πίνακες δρομολογίων.

αυθεντικό ρεαλιστικό χρονοεκτεταμένο, το ανηγμένο χρονοεκτεταμένο και για το γράφημα DTM, αντίστοιχα.

Ο συνδυασμός των αλγορίθμων απάντησης ερωτημάτων με την ευρετική τεχνική ALT απαιτεί μια φάση προεπεξεργασίας, για τον υπολογισμό των αποστάσεων όλων των κορυφών προς και από τα ορόσημα (Ενότητα 4.4). Οι υπολογιστικές απαιτήσεις αυτής της επεξεργασίας αποτυπώνονται στον Πίνακα 5.5.

Στο υπόλοιπο μέρος του κεφαλαίου παρουσιάζονται τα πειραματικά αποτελέσματα τόσο για τους χρόνους απόκρισης ερωτημάτων όσο και για την ενημέρωση των γραφοκεντρικών μοντέλων.

5.2 Ερωτήματα Πινάκων Δρομολογίων

Στην ενότητα αυτή παρουσιάζονται και αναλύονται τα πειραματικά αποτελέσματα των αλγορίθμων απόκρισης ερωτημάτων στους πίνακες δρομολογίων της Ενότητας 5.1.

Για την εξέταση της απόδοσης των τριών μοντέλων, εκτιμάται ο χρόνος που απαιτείται για την απάντηση ενός πλήθους ερωτημάτων NA για κάθε πίνακα δρομολογίων. Για κάθε πίνακα δρομολογίων, παράγονται 10,000 ερωτήσεις NA μεταξύ δύο σταθμών, των οποίων η επιλογή γίνεται με τυχαίο και διακριτό τρόπο. Στη συνέχεια υπολογίζεται ο χρόνος εκτέλεσης των αλγορίθμων για την απάντηση των ερωτημάτων σε κάθε γράφημα.

Οι αλγόριθμοι απάντησης ερωτημάτων που αξιολογούνται είναι: TE-Q, TE-QH, TE-QH-ALT, DTM-Q, DTM-QH και DTM-QH-ALT, αντίστοιχα.

Προκαταρκτικές πειραματικές αξιολογήσεις κατέδειξαν ότι η απόδοση του TE-QH είναι πάντα καλύτερη από εκείνη του TE-Q, και ότι ο αλγόριθμος DTM-QH είναι πάντα γρηγορότερος από τον DTM-Q. Για αυτούς τους λόγους, στον Πίνακα 5.6 αναφέρονται μόνο τα αποτελέσματα των TE-QH, TE-QH-ALT, DTM-QH και DTM-QH-ALT, αντίστοιχα.

Επιπλέον, έχουν εκτελεστεί ανάλογα πειράματα και για ερωτήματα ΕΑΜ, αλλά δεν παρουσιάζονται καθώς οδηγούν σε παρόμοια συμπεριφορά και ανάλυση.

Τύπος	$\mathcal{T}$	Ερώτημα (ms)			
		TE-QH-ALT	TE-QH	DTM-QH-ALT	DTM-QH
HaCon	bts	0.17	0.28	0.14	0.15
	efz	0.26	0.97	0.43	0.53
	ks	0.43	0.82	0.42	0.47
	bvb	1.02	1.77	1.51	1.79
	ger	1.21	4.48	1.53	4.08
	eur	1.61	5.13	1.58	4.32
GTFS	Βικτώρια	1.16	1.43	2.12	2.39
	Μαδρίτη	3.05	8.14	5.77	9.63
	Αθήνα	2.12	4.60	3.62	7.61
	Ρώμη	2.94	7.37	4.54	9.75
	Βερολίνο	6.88	18.34	12.17	27.63
	Λονδίνο	5.14	15.13	10.25	31.12
	Ελβετία	24.13	37.22	48.19	86.83
	Σουηδία	29.98	59.10	55.27	99.58

Πίνακας 5.6: Σύγκριση (μέσων) χρόνων απόκρισης ερωτημάτων μεταξύ των αλγορίθμων του ανηγμένου χρονοεκτεταμένου μοντέλου (TE-QH, TE-QH-ALT), και εκείνων του μοντέλου DTM (DTM-QH και DTM-QH-ALT).

Καταρχήν, τα πειραματικά αποτελέσματα (Πίνακας 5.6) κατέδειξαν ότι:

- Ο συνδυασμός της ευρετικής τεχνικής ALT με τα κριτήρια περιορισμού εξερεύνησης κορυφών (Ενότητες 4.1.2 και 4.2.4) οδηγεί σε σημαντική επιτάχυνση του χρόνου απάντησης ερωτημάτων σε σχέση με τους ήδη ενισχυμένους με ευρετικές τεχνικές αλγορίθμους TE-QH και DTM-QH για τα δύο μοντέλα
- Ο χρόνος απάντησης ερωτημάτων των αντίστοιχων αλγορίθμων του μοντέλου DTM είναι μεν υποδεέστερος, αλλά σχετικά ανταγωνίσimos με αυτόν

του ανηγμένου χρονοεκτεταμένου μοντέλου.

Όσον αφορά το δεύτερο αποτέλεσμα, σημειώνεται ότι μέρος της χρονικής επιβάρυνσης των αλγορίθμων απόκρισης ερωτημάτων του μοντέλου DTM πιθανά προκύπτει από το γεγονός ότι δεν υπάρχουν διακριτές ακμές άφιξης και αναχώρησης. Αυτό σημαίνει ότι κατά τη διάρκεια ενός ερωτήματος (S, T, t_S) , η εύρεση της επόμενης έγκυρης κορυφής αναχώρησης (δηλαδή της κορυφής που έχει χρόνο αναχώρησης μεγαλύτερο από t_S), πρέπει να αναζητηθεί μεταξύ πολλών (πιθανόν και όλων των) κορυφών αναχώρησης, αφού είναι αποθηκευμένες κατά αύξοντα χρόνο άφιξης στον επόμενο σταθμό, αντί του χρόνου αναχώρησης. Αυτό έχει σαν επακόλουθο την εκτέλεση των επιπλέον βημάτων της ενεργοποίησης ακμών και του υπολογισμού του κόστους τους. Το γεγονός αυτό κάνει τον αλγόριθμο πιο χρονοβόρο σε σχέση με αυτόν του ανηγμένου χρονοεκτεταμένου μοντέλου, όπου οι ακμές άφιξης και αναχώρησης επιτρέπουν την άμεση διεξαγωγή δυαδικής αναζήτησης εντός του συνόλου κορυφών αναχώρησης (ταξινομημένες κατά χρόνο αναχώρησης). Επομένως, η συρίκνωση των κορυφών άφιξης σε μια κορυφή (μεταγωγής) στο μοντέλο DTM είναι ένα μειονέκτημα αυτού του μοντέλου.

Άλλο ένα εξαγόμενο συμπέρασμα από την πειραματική αξιολόγηση είναι ότι όλοι οι αλγόριθμοι απόκρισης ερωτημάτων του μοντέλου DTM, δηλαδή οι DTM-Q, DTM-QH και DTM-QH-ALT, είναι πολύ γρηγορότεροι σε σχέση με προηγούμενες εκδοχές τους [19], αφού οι αλγόριθμοι αυτής της διατριβής είναι αποτέλεσμα περαιτέρω βελτιστοποίησης. Συγκρίνοντας αυτά τα αποτελέσματα με εκείνα που υπάρχουν στον Πίνακα 5.6, είναι εύκολο να διαπιστώσει κανείς ότι: (α) η νέα έκδοση του DTM-QH είναι έως και 50% γρηγορότερη σε σχέση με την προηγούμενη· (β) η νέα έκδοση του DTM-QH-ALT είναι έως 30-35% γρηγορότερη από την προηγούμενη. Επιπρόσθετα, επειδή είναι γνωστό ότι ο χρόνος απόκρισης ερωτημάτων στα χρονοεξαρτώμενα γραφήματα είναι γρηγορότερος από εκείνον των χρονοεκτεταμένων κατά ένα μικρό καθορισμένο σταθερό παράγοντα [72], τα πειραματικά αποτελέσματα της παρούσας διδακτορικής διατριβής έδειξαν ότι ο χρόνος απόκρισης ερωτημάτων του μοντέλου DTM είναι συγκρίσιμος με τον αντίστοιχο χρόνο του χρονοεξαρτώμενου μοντέλου.

Ο χρόνος απόκρισης ερωτημάτων εξαρτάται επίσης από κάποια σημαντικά

χαρακτηριστικά των συστημάτων μέσω μαζικής μεταφοράς (MMM), όπως είναι ο αριθμός των γειτονικών στάσεων/σταθμών και το πλήθος των διαφορετικών ειδών MMM. Με βάση τους Πίνακες 5.3 και 5.4, τα δίκτυα MMM της Σουηδίας, Ελβετίας και Βερολίνου έχουν μεγαλύτερη πυκνότητα σε αυτά τα χαρακτηριστικά, με αποτέλεσμα να είναι πιο χρονοβόρος ο υπολογισμός των βέλτιστων διαδρομών σε σύγκριση με τα υπόλοιπα δίκτυα MMM.

Τέλος, τα πειραματικά αποτελέσματα καταδεικνύουν ότι ο χρόνος απόκρισης των γραφοκεντρικών μοντέλων είναι συγκρίσιμος σε σχέση με τους αντίστοιχους χρόνους των διανυσματοκεντρικών μοντέλων. Για παράδειγμα, ο χρόνος απόκρισης του αλγορίθμου CSA είναι 2ms σε ένα στιγμιότυπο του μητροπολιτικού δικτύου του Λονδίνου με περίπου 5 εκατομμύρια συνδέσεις [6], ενώ ο χρόνος απόκρισης ερωτημάτων των αλγορίθμων TE-QH-ALT και DTM-QH-ALT είναι 5.14ms και 10.25ms, αντίστοιχα, σε ένα στιγμιότυπο του μητροπολιτικού δικτύου του Λονδίνου με 14 εκατομμύρια συνδέσεις (δηλαδή σχεδόν τρεις φορές μεγαλύτερο).

5.3 Ενημέρωση Πινάκων Δρομολογίων

Στην ενότητα αυτή παρουσιάζονται και αναλύονται τα πειραματικά αποτελέσματα των αλγορίθμων ενημέρωσης των πινάκων δρομολογίων της Ενότητας 5.1 μετά από μια καθυστέρηση.

Για την αξιολόγηση της απόδοσης των αλγορίθμων ενημέρωσης, εκτελέστηκε ένα σύνολο πειραμάτων ως εξής: για κάθε πίνακα δρομολογίων έγινε τυχαία επιλογή 1,000 στοιχειωδών συνδέσεων, και για κάθε στοιχειώδη σύνδεση έγινε τυχαία παραγωγή μίας καθυστέρησης που επηρεάζει το αντίστοιχο τρένο ή λεοφορείο, της οποίας η τιμή επιλέχθηκε στο διάστημα $[1, 360]$ λεπτών με βάση την ομοιόμορφη κατανομή πιθανότητας.

Για κάθε αλλαγή του πίνακα δρομολογίων, εκτελέστηκε ο αλγόριθμος DTM-U για την ενημέρωση της πληροφορίας του γραφήματος DTM (Ενότητα 4.2.3), καθώς επίσης και ο αλγόριθμος TE-UH για την ενημέρωση της πληροφορίας του ανηγμένου χρονοεκτεταμένου γραφήματος (Ενότητα 4.1.3). Μετρήθηκε ο μέσος χρόνος υπολογισμού και ο αριθμός των συνδέσεων που επηρεάζονται

Τύπος	$\mathcal{T}$	Ενημέρωση (μs)			
		TE-UH		DTM-U	
		Χρόνος	# ακμών	Χρόνος	# ακμών
HaCon	bts	34.47	37.7	2.19	8.8
	efz	25.31	30.9	2.25	7.2
	ks	39.49	53.8	3.04	11.1
	bvb	95.98	63.5	15.17	13.5
	ger	40.32	32.2	7.17	7.6
	eur	43.10	33.5	8.07	9.1
GTFS	Βικτώρια	96.36	64.7	14.31	10.2
	Μαδρίτη	124.50	64.5	29.45	11.1
	Αθήνα	424.74	181.0	46.98	59.4
	Ρώμη	381.47	107.3	49.73	81.4
	Βερολίνο	194.54	61.6	80.23	9.9
	Λονδίνο	477.23	130.4	271.46	29.0
	Ελβετία	678.92	55.5	356.15	40.3
	Σουηδία	604.77	85.5	293.62	20.0

Πίνακας 5.7: Σύγκριση (μέσων) χρόνων ενημέρωσης δομής γραφημάτων μεταξύ του αλγορίθμου TE-UH του ανηγμένου χρονοεκτεταμένου μοντέλου, και του DTM-U του μοντέλου DTM. Καταγράφεται επίσης ο μέσος αριθμός των ακμών που επηρεάζονται, δηλαδή, οι ακμές που ενημερώνονται (με αλλαγή στο κόστος/ή επανατοποθέτηση) από τον TE-UH, καθώς και οι στοιχειώδεις συνδέσεις που απαιτούν ενημέρωση από τον DTM-U.

από την καθυστέρηση, δηλαδή ο αριθμός των συνδέσεων που συσχετίζονται με το ίδιο τρένο ή λεωφορείο που έχει υποστεί την καθυστέρηση. Παραλείπονται τα αποτελέσματα του αλγορίθμου ενημέρωσης του αυθεντικού ρεαλιστικού χρονοεκτεταμένου γραφήματος TE-U, καθώς τα αρχικά πειράματα κατέδειξαν ότι ο αλγόριθμος ενημέρωσης του ανηγμένου χρονοεκτεταμένου γραφήματος TE-UH ήταν πάντοτε τουλάχιστον 50% γρηγορότερος από τον TE-U.

Τα πειραματικά αποτελέσματα καταγράφονται στον Πίνακα 5.7. Ο αλγόριθμος DTM-U υπερέρχει σε σχέση με τον TE-UH. Τα αποτελέσματα επιβεβαιώνουν ότι το άνω όριο της χρονικής πολυπλοκότητας του αλγορίθμου DTM-U (Θεώρημα 9) έχει πολύ μεγάλη απόκλιση από την πραγματικότητα και επομένως, το μοντέλο DTM είναι κατάλληλο για πρακτική εφαρμογή. Στην πραγματικότητα, ακόμη και στα μεγαλύτερα δίκτυα (Λονδίνου, Σουηδίας και Ελβετίας) ο χρόνος του αλγορίθμου ενημέρωσης είναι λιγότερο από 360 μs . Επιπλέον, μόνο το πολύ δύο κόστη ακμών και λίγες χρονοσφραγίδες κορυφών πρέπει να ενημερωθούν στο αρχικό γράφημα για να συνεχίσουν να απαντώνται σωστά τα

ερωτήματα NA. Στους πίνακες δρομολογίων που εξετάστηκαν, μετρήθηκε ένας μέγιστος (κατά μέσο όρο) αριθμός 81.4 στοιχειωδών συνδέσεων που είχαν επηρεαστεί. Αυτό οφείλεται στο γεγονός ότι ο αριθμός των σταθμών στους οποίους συμβαίνουν αλλαγές, ως συνέπεια της καθυστέρησης, είναι μικρός σε σχέση με το μέγεθος $|B|$ του συνόλου όλων των σταθμών B .

5.4 Σύγκριση με τον RAPTOR

Για λόγους πληρότητας, έγινε απευθείας σύγκριση του χρόνου απόκρισης των γραφοκεντρικών μοντέλων που αναπτύχθηκαν στην παρούσα διδακτορική διατριβή, με έναν από τους γρηγορότερους διανυσματοκεντρικούς αλγορίθμους που δεν στηρίζεται σε φάση προεπεξεργασίας, ονομάζεται Round-Based Public Transit Routing (RAPTOR), και παρουσιάζεται στις εργασίες [28, 29].

Ο RAPTOR έχει αποκλειστικά αναπτυχθεί για δημόσια δίκτυα μεταφορών και στη βασική του έκδοση βελτιστοποιεί τον χρόνο άφιξης και τον αριθμό των μετεπιβάσεων που απαιτούνται. Αντί να χρησιμοποιεί ένα γράφημα, οργανώνει την είσοδο ως μερικά απλά διανύσματα ταξιδιών και διαδρομών. Σημειώνεται ότι ένα *ταξίδι* (trip) είναι μια ακολουθία σταθμών που συνδέονται με στοιχειώδεις συνδέσεις οι οποίες χρησιμοποιούνται από το ίδιο όχημα, ενώ μια *διαδρομή* (route) είναι μια σειρά ταξιδιών που διαμοιράζονται το ίδιο σύνολο σταθμών.

Ουσιαστικά, ο RAPTOR είναι ένα δυναμικό πρόγραμμα: λειτουργεί σε επανάλψεις, όπου στην επανάληψη i υπολογίζονται οι χρόνοι NA για διαδρομές που αποτελούνται από ακριβώς i μετεπιβιώσεις. Κάθε επανάληψη παίρνει ως είσοδο τις στάσεις των οποίων ο χρόνος άφιξης βελτιώθηκε στην προηγούμενη επανάληψη (για την πρώτη επανάληψη αυτή είναι μόνο η στάση της αφετηρίας). Στη συνέχεια σαρώνει τις διαδρομές που εξυπηρετούνται από αυτές τις στάσεις. Για να σαρωθεί η διαδρομή r , ο RAPTOR διατρέχει τις στάσεις κατά σειρά ταξιδιών, παρακολουθώντας το νωρίτερο δυνατό ταξίδι (της r) που μπορεί να χρησιμοποιηθεί. Αυτό το ταξίδι μπορεί να βελτιώσει τους προσωρινούς χρόνους άφιξης στις επόμενες στάσεις της διαδρομής r . Σημειώνεται επίσης ότι ο RAPTOR σαρώνει κάθε διαδρομή το πολύ μία φορά ανά επανάληψη, κάτι το οποίο είναι πολύ αποτελεσματικό στην πράξη. Επιπλέον, ο RAPTOR μπορεί να

παραλληλοποιηθεί κατανέμοντας μη-τεμνόμενες διαδρομές σε διαφορετικούς πυρήνες επεξεργαστών. Περισσότερες πληροφορίες σχετικά με τον αλγόριθμο και την απόδοσή του περιγράφονται στην εργασία [29].

	TE-QH	TE-QH-ALT	DTM-QH	DTM-QH-ALT	RAPTOR
Ερώτημα (ms)	15.13	5.14	31.12	10.25	7.79

Πίνακας 5.8: Οι χρόνοι απόκρισης ερωτημάτων μεταξύ των γραφοκεντρικών μοντέλων και του RAPTOR στο δημόσιο δίκτυο MMM του Λονδίνου (περίπου 14M στοιχειώδεις συνδέσεις).

Η σύγκριση μεταξύ των γραφοκεντρικών μοντέλων και του RAPTOR αποτυπώνεται στον Πίνακα 5.8. Συγκεκριμένα, καταγράφεται για κάθε μοντέλο ο μέσος χρόνος απόκρισης ερωτημάτων σε ένα στιγμιότυπο του μητροπολιτικού συστήματος MMM του Λονδίνου, το οποίο αποτελείται από περίπου 14M στοιχειώδεις συνδέσεις.

Δυστυχώς, δεν είναι εφικτή η απευθείας αξιολόγηση του RAPTOR στην περίπτωση της ενημέρωσης (λόγω καθυστερήσεων). Για την απόδοσή του στο δυναμικό σενάριο, ούτε υπάρχει αναφορά στην βιβλιογραφία, ούτε είναι ξεκάθαρο πώς μπορεί να εφαρμοστεί ή να τροποποιηθεί ο κώδικας του RAPTOR ώστε να διαχειρίζεται τις καθυστερήσεις στους πίνακες δρομολογίων. Συγκεκριμένα, ο RAPTOR, για να εκμεταλλευτεί την τοπικότητα αναφοράς δεδομένων, ταξινομεί τόσο τις διαδρομές όσο και τα ταξίδια σε διανύσματα σύμφωνα με το χρόνο [29]. Επομένως, για να διατηρηθεί η ορθότητα απάντησης ερωτημάτων, θα έπρεπε μετά από κάθε ενημέρωση του αρχικού πίνακα δρομολογίων: (α) να ενημερώνεται ο χρόνος που συσχετίζεται με ταξίδια και διαδρομές· (β) να ελέγχεται αν μετά την ενημέρωση παραβιάζεται η διάταξη των διανυσμάτων, και αν όντως, τότε να μετακινούνται αναλόγως εντός των διανυσμάτων ώστε να αποκαθίσταται η διάταξη.

Στον Πίνακα 5.8 καταδεικνύεται ότι τα γραφοκεντρικά μοντέλα που αναπτύχθηκαν στα πλαίσια της παρούσας διδακτορικής διατριβής είναι συγκρίσιμα τον RAPTOR όσον αφορά το χρόνο απόκρισης ερωτημάτων. Επιπλέον, καταδεικνύεται ότι ο αλγόριθμος απάντησης ερωτημάτων του ανηγμένου χρονοεκτεταμένου μοντέλου, όταν συνδυαστεί με τον ALT, είναι γρηγορότερος από εκείνον του RAPTOR.

Σημειώνεται ότι όλα τα γραφοκεντρικά μοντέλα θεωρούνται στη γενική κυκλική (περιοδική) τους έκδοση, δηλαδή, δεν εκτυλίσσεται ο πίνακας δρομολογίων σε πολλές συνεχόμενες περιόδους προκειμένου να προκύψουν ακυκλικοί και απεριοδικοί πίνακες δρομολογίων και να αντιμετωπιστούν μεταμεσονύκτιες συνδέσεις, όπως απαιτείται από τα διανυσματοκεντρικά μοντέλα. Για το λόγο αυτό, στα γραφοκεντρικά μοντέλα πρέπει να χρησιμοποιηθεί ένας αλγόριθμος βασισμένος σε μια ουρά προτεραιότητας για τον υπολογισμό των βέλτιστων διαδρομών, όπως οι τροποποιήσεις του αλγορίθμου Dijkstra που περιγράψαμε στο Κεφάλαιο 4. Αυτή η επιλογή επιτρέπει την εξοικονόμηση χώρου με τίμημα μια πολύ μικρή υστέρηση στο χρόνο απόκρισης ερωτημάτων. Μια διαφορετική αντιστάθμιση θα είχε ως αποτέλεσμα έναν καλύτερο χρόνο απόκρισης ερωτημάτων με τίμημα μια επιβάρυνση στο χώρο αποθήκευσης των πινάκων δρομολογίων.

Κεφάλαιο 6

Εφαρμογή Κινητών Συσκευών

Στο κεφάλαιο αυτό παρουσιάζεται μια εφαρμογή ψηφιακού ταξιδιωτικού πλοηγού (journey planner) μέσω μιας μαζικής μεταφοράς, η οποία λειτουργεί σε έξυπνες κινητές συσκευές με λειτουργικό σύστημα Android. Ο ψηφιακός ταξιδιωτικός πλοηγός βασίζεται σε υποδομή υπολογιστικού νέφους και περιέχει μία αποδοτική αλγοριθμική μηχανή δρομολόγησης η οποία βασίζεται στους αποδοτικούς αλγόριθμους εύρεσης βέλτιστων διαδρομών και ενημέρωσης πληροφορίας δρομολογίων που παρουσιάστηκαν στο Κεφάλαιο 4. Στο Κεφάλαιο αυτό παρουσιάζεται επίσης το περιβάλλον υλοποίησης του ψηφιακού ταξιδιωτικού πλοηγού και τα χαρακτηριστικά του, τα οποία περιλαμβάνουν (μεταξύ άλλων) και μια υπηρεσία αξιολόγησης των προτεινόμενων διαδρομών από τους χρήστες.

6.1 Εισαγωγή

Στη σημερινή εποχή της διαρκώς αυξανόμενης κινητικότητας πολιτών (και αγαθών) παρατηρείται μια πληθώρα εφαρμογών κινητών συσκευών και διαδικτύου, γνωστών και ως ψηφιακών *ταξιδιωτικών πλοηγών*, οι οποίοι υπολογίζουν καλύτερες/βέλτιστες διαδρομές σε συστήματα δημόσιας συγκοινωνίας. Δεδομένου ενός πίνακα δρομολογίων, που συσχετίζεται με ένα δημόσιο σύστημα μεταφορών, το πρόβλημα της *ταξιδιωτικής πλοήγησης* (journey planning) αφορά στην αποδοτική απάντηση ερωτημάτων της μορφής: «Ποια είναι η καλύτερη

διαδρομή από κάποιον σταθμό A σε κάποιον άλλο σταθμό B , υπό την προϋπόθεση ότι θέλω να αναχωρήσω από τον σταθμό A την χρονική στιγμή t ;».

Όπως ήδη έχει περιγραφεί στο Κεφάλαιο 3, το πρόβλημα της ταξιδιωτικής πλοήγησης (εύρεσης βέλτιστων διαδρομών υπό κάποια μετρική) εξειδικεύεται σε διάφορα προβλήματα βελτιστοποίησης, εκ των οποίων τα δύο πιο θεμελιώδη αφορούν το πρόβλημα της νωρίτερης άφιξης (NA) και το πρόβλημα του ελάχιστου αριθμού μετεπιβιβάσεων (EAM). Τα προβλήματα αυτά ορίσθηκαν στην Ενότητα 3.1. Υπενθυμίζεται ότι η τυπική αντιμετώπιση τέτοιων προβλημάτων είναι η δημιουργία, σε μια φάση προεπεξεργασίας, μιας δομής δεδομένων για την αναπαράσταση του πίνακα δρομολογίων, έτσι ώστε ερωτήματα εύρεσης βέλτιστων διαδρομών να απαντώνται ταχύτατα.

Το πρόβλημα εύρεσης βέλτιστων διαδρομών με χρήση μέσων μαζικής μεταφοράς είναι αρκετά περίπλοκο (σε αντίθεση με την απλή του διατύπωση), πολύ περισσότερο από το αντίστοιχο πρόβλημα στα οδικά δίκτυα. Τα συστήματα μεταφορών βασισμένα σε πίνακες δρομολογίων απαιτούν, λόγω της χρονοεξαρτώμενης φύσης τους, μοντελοποίηση με περισσότερο πολύπλοκες παραδοχές (Κεφάλαια 3 και 4) προκειμένου να παράγουν λογικά και χρήσιμα αποτελέσματα, ιδίως όταν πρέπει να ληφθούν υπόψη και οι χρόνοι μετεπιβίβασης από ένα όχημα σε ένα άλλο.

Μία επιπλέον πρόκληση, όπως περιγράφηκε και στα Κεφάλαια 3 και 4, είναι η αποδοτική υποστήριξη των καθυστερήσεων που συχνά συμβαίνουν στα οχήματα των μέσω μαζικής μεταφοράς. Το κεντρικό ζήτημα είναι η αποδοτική ενημέρωση της δομής των πινάκων δρομολογίων, έτσι ώστε η ανάκτηση της καλύτερης διαδρομής (συνήθως νωρίτερης άφιξης) να μπορεί να γίνει γρήγορα και βέλτιστα σε σχέση με τον αναθεωρημένο πίνακα δρομολογίων.

Στην πράξη, ένας ταξιδιωτικός πλοηγός είναι συνήθως σε πολύ μεγάλη ζήτηση. Για παράδειγμα, ο ταξιδιωτικός πλοηγός των Γερμανικών σιδηροδρόμων λαμβάνει σε ώρες αιχμής περισσότερα από 420 ερωτήματα εύρεσης βέλτιστων διαδρομών ανά δευτερόλεπτο. Το γεγονός αυτό, σε συνδυασμό με το ότι καθυστερήσεις συμβαίνουν συχνά, απαιτούν την υποστήριξη μιας ιδιαίτερα απαιτητικής υπολογιστικής υπηρεσίας, η οποία χρειάζεται μια αρχιτεκτονική επαρκούς υπολογιστικής ικανότητας. Από την άποψη αυτή, η ελαστικότητα μιας αρχιτεκτονικής υπολογιστικού νέφους επιτρέπει την προσαρμογή των δεσμευ-

μένων υπολογιστικών πόρων στις πραγματικές απαιτήσεις επεξεργασίας ενός μεγάλου αριθμού ερωτημάτων και ενημέρωσης της πληροφορίας ενός πίνακα δρομολογίων.

Το *δυναμικό μοντέλο πινάκων δρομολογίων* (DTM) που αναπτύχθηκε στα πλαίσια αυτής της διδακτορικής διατριβής και παρουσιάστηκε αναλυτικά στο Κεφάλαιο 4 αποτελεί ένα γραφοκεντρικό χωρο-αποδοτικό μοντέλο αναπαράστασης πληροφορίας πινάκων δρομολογίων. Διαθέτει έναν αποδοτικό αλγόριθμο απάντησης ερωτημάτων νωρίτερης άφιξης καθώς και τον ταχύτερο αλγόριθμο ενημέρωσης της πληροφορίας του πίνακα δρομολογίων, όταν συμβαίνουν καθυστερήσεις. Όπως διαπιστώθηκε και στο Κεφάλαιο 5, τα πειραματικά αποτελέσματα σε πραγματικά συγκοινωνιακά δεδομένα κατέδειξαν την εξαιρετική πρακτική απόδοση του μοντέλου DTM.

Οι προαναφερθείσες προκλήσεις για την απόκριση σε πραγματικό χρόνο και την αποτελεσματική ενσωμάτωση καθυστερήσεων σε ένα σύστημα πληροφοριών πινάκων δρομολογίων απαιτούν την υιοθέτηση πιο προηγμένων αρχιτεκτονικών. Η καρδιά μιας τέτοιας προηγμένης υπηρεσίας ταξιδιωτικού πλοηγού πρέπει να βρίσκεται σε μια αρχιτεκτονική υπολογιστικού νέφους, η οποία είναι σε θέση να εγγυηθεί τη διαρκή διατήρηση των δεδομένων, τη διαλειτουργικότητα με άλλες πηγές (που αναφέρουν καθυστερήσεις οχημάτων), την ελαστικότητα των υπολογιστικών πόρων σε πραγματικό χρόνο, καθώς και τη διάφανη προσβασιμότητα των ταξιδιωτών στον πλοηγό. Σε ένα τέτοιο περιβάλλον, τα ερωτήματα αποστέλλονται σε μια «μηχανή δρομολόγησης» (routing engine) που βρίσκεται στην υποδομή του υπολογιστικού νέφους, η οποία με τη σειρά της στέλνει πίσω τις απαντήσεις λαμβάνοντας υπόψη την ενημερωμένη πληροφορία του πίνακα δρομολογίων.

Στο Κεφάλαιο αυτό παρουσιάζεται ένας ψηφιακός ταξιδιωτικός πλοηγός MMM βασισμένος σε υποδομή υπολογιστικού νέφους, του οποίου η κεντρική μηχανή δρομολόγησης βασίζεται στο δυναμικό μοντέλο πινάκων δρομολογίων DTM. Παρουσιάζεται η εφαρμογή κινητών συσκευών (mobile application) για τον ταξιδιώτη, μαζί με μια υπηρεσία που επιτρέπει στους χρήστες να αξιολογήσουν τις προτεινόμενες διαδρομές που προσφέρει η εφαρμογή. Η υπηρεσία αξιολόγησης χρηστών έχει βασιστεί σε μια υποδομή «διαδικτύου αντικειμένων» (Internet of Things), συγκεκριμένα την υποδομή Organicity [71], η οποία α-

ναπτύχθηκε στα πλαίσια του αντίστοιχου ερευνητικού έργου του Ευρωπαϊκού Προγράμματος IoT/FIRE+.

Ο ταξιδιωτικός πλοηγός συμβάλλει στη δημιουργία μιας ζωντανής κοινότητας ταξιδιωτών, εξοπλισμένων με εργαλεία διαλειτουργικών εξατομικευμένων υπηρεσιών ανανεωνόμενης κινητικότητας, για σύγχρονη κινητικότητα σε έξυπνες πόλεις όπως αυτή προδιαγράφηκε στα πλαίσια του έργου [65].

6.2 Το Περιβάλλον Ανάπτυξης του Ταξιδιωτικού Πλοηγού

Η διεπαφή χρήστη του ταξιδιωτικού πλοηγού αναπτύχθηκε σε περιβάλλον Android. Η επικοινωνία της διεπαφής χρήστη με τη μηχανή δρομολόγησης γίνεται με ανταλλαγή μηνυμάτων σε μορφή JSON, ενώ η υπηρεσία αξιολόγησης διαδρομών χρησιμοποιεί το πλαίσιο του Πειραματισμού-ως-Υπηρεσία (Experimentation-as-a-Service) που παρέχει η πλατφόρμα Organicity. Σε αυτή την ενότητα παρουσιάζεται το περιβάλλον ανάπτυξης της εφαρμογής του ταξιδιωτικού πλοηγού.

6.2.1 Το Περιβάλλον Κινητών Συσκευών

Το Android [3] είναι το πλέον διαδεδομένο λειτουργικό σύστημα κινητών συσκευών, το οποίο τρέχει τον πυρήνα του λειτουργικού συστήματος Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance [70]. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα έξυπνα τηλέφωνα και τα τάμπλετ, ενώ έχει χρησιμοποιηθεί και σε άλλες συσκευές με διαφορετικό περιβάλλον χρήσης (π.χ. τηλεοράσεις, αυτοκίνητα, ρολόγια χειρός, ψηφιακές φωτογραφικές μηχανές, κονσόλες παιχνιδιών). Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της ελεύθερης άδειας λογισμικού Apache. Το λειτουργικό σύστημα Android διαθέτει τα απα-

ραίτητα πακέτα για χρήση όλων των λειτουργιών της υπηρεσίας GPS, που είναι απαραίτητες για την ανάπτυξη του ταξιδιωτικού πλοηγού.

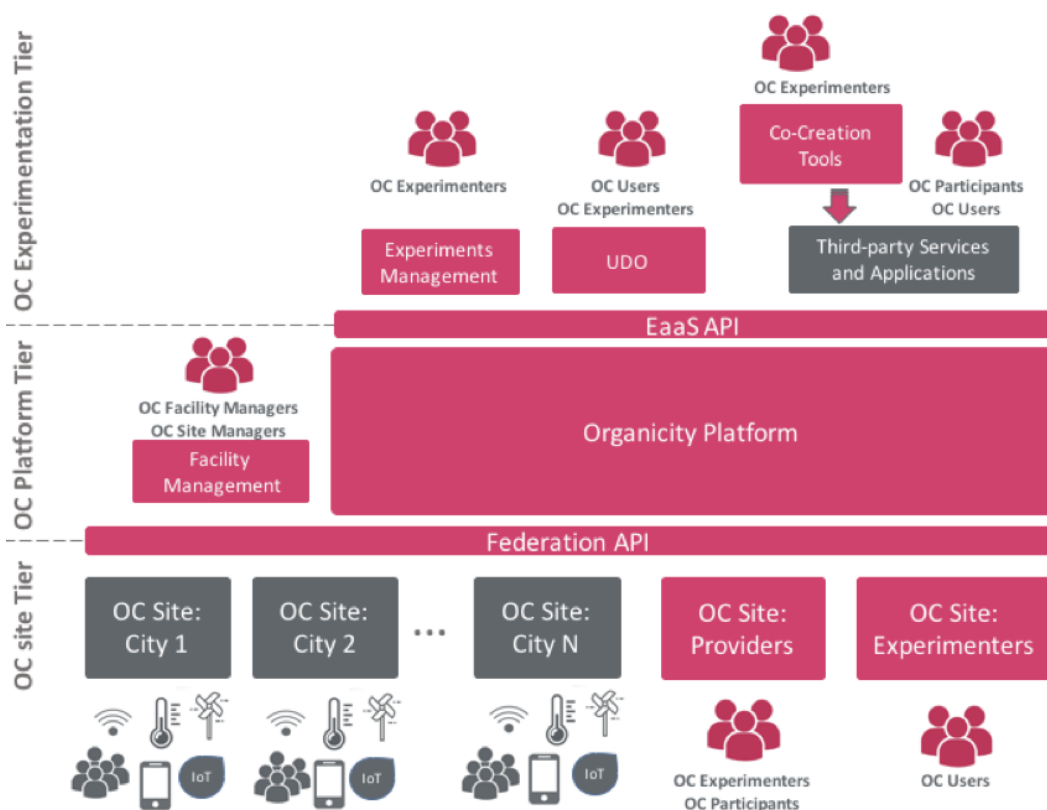
Η διεπαφή χρήστη του ταξιδιωτικού οδηγού αναπτύχθηκε στο περιβάλλον Android Studio [4], που είναι το επίσημο ολοκληρωμένο αναπτυξιακό περιβάλλον (Integrated Development Environment – IDE) της πλατφόρμας Android. Το Android Studio δίνει την δυνατότητα σταδιακής ανάπτυξης εφαρμογών, με μία λίστα πολλών έτοιμων εργαλείων και έναν εξομοιωτή (emulator) που εμφανίζει το αποτέλεσμα της εφαρμογής που έχει αναπτυχθεί. Έτσι ο προγραμματιστής δημιουργεί την τελική μορφή της εφαρμογής του και ταυτόχρονα δημιουργείται αυτόματα και ο κώδικας XML, στον οποίο ο προγραμματιστής έχει την δυνατότητα να παρέμβει. Επιπλέον, εκτός από το σχεδιαστικό μέρος της διεπαφής, το Android Studio δίνει την δυνατότητα προγραμματισμού και με τις συμβατές με την XML γλώσσες προγραμματισμού java, javascript, php για την περαιτέρω δημιουργία πρόσθετων λειτουργιών.

6.2.2 Η Πλατφόρμα Organicity

Η πλατφόρμα Organicity (OC) [71] παρέχει ένα πλαίσιο αξιολόγησης η οποία υποστηρίζει εκτός των άλλων, την κατασκευή υπηρεσιών πειραματισμού. Η αρχιτεκτονική της πλατφόρμας σε υψηλό επίπεδο παρουσιάζεται στο Σχήμα 6.1.

Το Organicity είναι ένα εν εξελίξει ερευνητικό έργο που χρηματοδοτείται από την Ευρωπαϊκή Ένωση (στα πλαίσια του Ευρωπαϊκού Προγράμματος I-οT/FIRE+). Δημιουργεί πλατφόρμες αξιολόγησης οι οποίες θα υποστηρίξουν πειράματα με την αξιοποίηση τεχνολογιών διαδικτύου αντικειμένων (Internet Of Things – IoT) και θα παρέχουν υπηρεσίες έξυπνων πόλεων. Παραδείγματα προτεινόμενων σεναρίων χρήσης περιλαμβάνουν παρακολούθηση θέσεων στάθμευσης αυτοκινήτων, παρακολούθηση περιβαλλοντολογικών μετρήσεων, καθώς και χρήση ετικετών τύπου Near Field Communication (NFC). Οι τελικοί χρήστες αυτής της πλατφόρμας αναμένονται να είναι μεταξύ των ακόλουθων κατηγοριών:

- Ομάδες πολιτών που ενδιαφέρονται να συνεισφέρουν δεδομένα ή ιδέες ή να αναπτύξουν οι ίδιοι νέες εφαρμογές σε ένα περιβάλλον έξυπνης πόλης.



Σχήμα 6.1: Η υψηλού επιπέδου αρχιτεκτονική της πλατφόρμας Organicity.

- Ερευνητές από διάφορους κλάδους, οι οποίοι επιθυμούν να εκτελέσουν πειράματα IoT χρησιμοποιώντας τις υπηρεσίες που προσφέρει η πλατφόρμα Organicity.
- Εμπορικοί φορείς (εταιρείες, μικρο-μεσαίες επιχειρήσεις, κλπ.) που επιθυμούν να δοκιμάσουν καινοτόμες ιδέες στη νέα πλατφόρμα χρησιμοποιώντας την παρεχόμενη υποδομή ή έχοντας, πρόσβαση σε σύνολα δεδομένων.
- Αρχές/φορείς πόλεων που επιθυμούν να αναπτύξουν νέες επωφελείς υπηρεσίες για τη δική τους λειτουργία και να παρέχουν υψηλότερα επίπεδα πληροφόρησης στους πολίτες τους.

Αυτό το νέο μοντέλο έξυπνων πόλεων προορίζεται να χρησιμεύσει ως εκκολλητήριο ανάπτυξης ενός ποικίλου συνόλου ιδιαίτερα καινοτόμων υπηρεσιών

και εφαρμογών. Η δημιουργία μιας πειραματικής υπηρεσίας μέσα σε μια πόλη σημαίνει την παροχή συνθηκών πραγματικού κόσμου μαζί με την απαραίτητη υποδομή έξυπνων πόλεων που βασίζεται σε τεχνολογίες IoT. Η κλίμακα και η ετερογένεια ενός τέτοιου περιβάλλοντος καθιστούν αυτή την ανάπτυξη εξαιρετικά πολύτιμη για τους σκοπούς της επιστημονικής και τεχνικής έρευνας. Αυτό το πλαίσιο μεγέθους πόλης αποσκοπεί στην προσέλκυση του απαραίτητου πλήθους πειραματιζόμενων επιχειρήσεων και τελικών χρηστών που απαιτούνται για τη δοκιμή τεχνολογιών IoT σε μεγάλη κλίμακα, καθώς επίσης και στο συνδυασμό αυτών με άλλες μελλοντικές τεχνολογίες διαδικτύου που η αγορά θα υιοθετήσει.

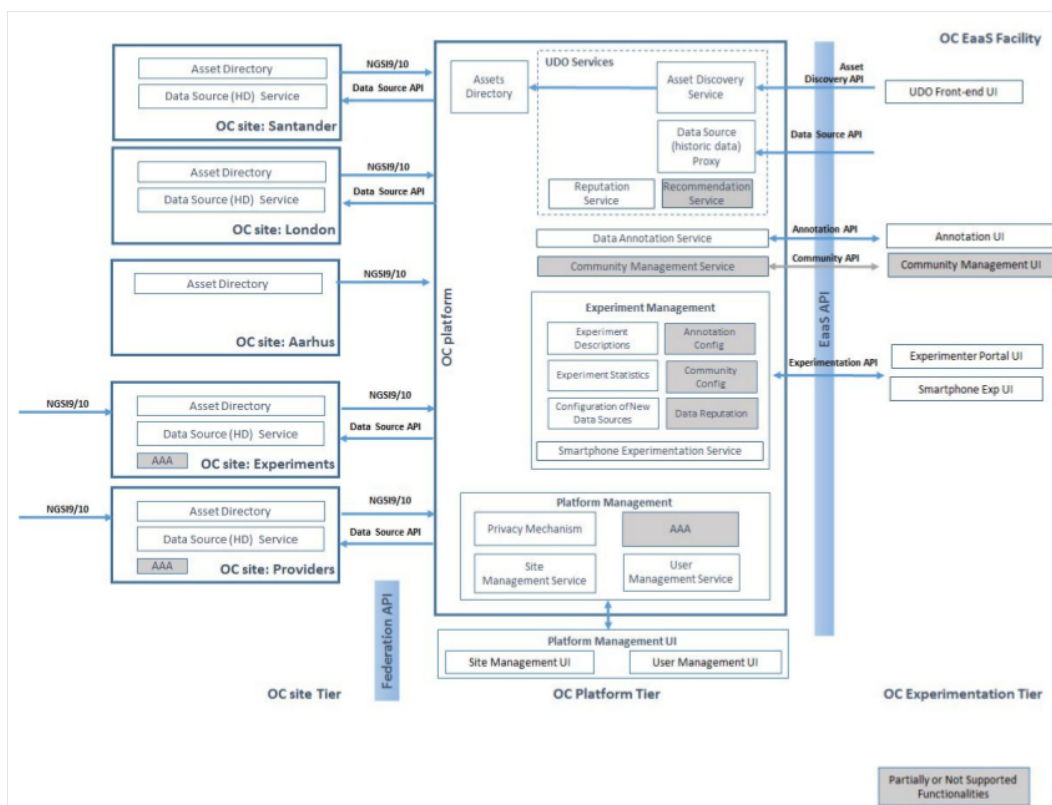
Η πλατφόρμα *Organicity* παρέχει μια ολιστική προσέγγιση που βασίζεται στον πολίτη για τη βελτιστοποίηση της πόλης συνδυάζοντας τον φυσικό με τον ψηφιακό χώρο. Μια βασική πτυχή των υπηρεσιών της πλατφόρμας *Organicity* είναι ότι οι πολίτες δεν διαδραματίζουν μόνο το ρόλο «παραγωγών δεδομένων», παρέχοντας δεδομένα σχετικά με την πόλη μέσω της συμμετοχικής ανίχνευσης (*participatory sensing*), του πληθοπορισμού (*crowdsourcing*) και των κοινωνικών δικτύων. Αλλά, θα διαδραματίσουν επιπλέον και το ρόλο του «καταναλωτή δεδομένων» αναπτύσσοντας υπηρεσίες πάνω σε αυτά τα δεδομένα, ή επεξεργάζοντας σύνολα δεδομένων και σχολιάζοντάς τα, παρέχοντας έτσι γνώση υψηλού επιπέδου μέσω των μοναδικών διερμηνεύσεών τους.

Η υλοποίηση του πλαισίου *Πειραματισμός-ως-Υπηρεσία* (*Experimentation-as-a-Service*) της πλατφόρμας *Organicity* είναι μια επαναληπτική, ευέλικτη και προσαρμοστική προσέγγιση στις μεταβαλλόμενες ανάγκες διαφόρων ομάδων τελικών χρηστών που συνεργάζονται με την πλατφόρμα. Το έργο *Organicity* παρέχει ένα κοινό πλαίσιο ενοποιημένης πλατφόρμας πειραμάτων, σε σχέση με τις διαφορετικές φυσικές πλατφόρμες αξιολόγησης — στο έργο συμμετέχουν τρεις βασικές συμβεβλημένες τοποθεσίες (*Aarhus*, *London* και *Santander*) που προσφέρουν κάποιες αρχικές πλατφόρμες εκτέλεσης πειραμάτων.

Το πλαίσιο Πειραματισμός-ως-Υπηρεσία (*EaaS*) του *Organicity* καλύπτει την τεχνική ετερογένεια συγκεκριμένων υποκείμενων υποδομών πίσω από τις γενικές προγραμματιστικές διεπαφές εφαρμογών (*Application Programming Interfaces – APIs*), επιτρέποντας στους πειραματιστές (ερευνητές, προγραμμα-

τιστές υπηρεσιών αστικών περιβαλλόντων, πολίτες κ.λπ.) να έχουν πρόσβαση σε διαθέσιμες πηγές δεδομένων, επιτρέποντας τη συν-δημιουργία νέων υπηρεσιών και εφαρμογών χρησιμοποιώντας διαφορετικά εργαλεία. Επιπλέον, η πλατφόρμα επιτρέπει να συμπεριληφθούν και άλλα «αστικά ζωντανά εργαστήρια» (city living labs) στο πλαίσιο του έργου Organicity.

Σε αυτήν την ενότητα, παρουσιάζονται οι βασικές λειτουργίες των υπηρεσιών και των διεπαφών της πλατφόρμας Organicity που είναι ήδη διαθέσιμες στους τελικούς χρήστες: συμβεβλημένες τοποθεσίες (sites), πειραματιστές, συμμετέχοντες σε ένα πείραμα, ή απλοί χρήστες.



Σχήμα 6.2: Το πλαίσιο Πειραματισμού-ως-Υπηρεσία (Experimentation-as-a-Service – EaaS Facility) της πλατφόρμας Organicity.

Στο Σχήμα 6.2 παρουσιάζεται μια υψηλού επιπέδου επισκόπηση των διαθέσιμων συστατικών τμημάτων λογισμικού, των λειτουργιών, και των διεπαφών (APIs του πλαισίου Πειραματισμός-ως-Υπηρεσία (EaaS) της πλατφόρμας Or-

ganicity. Στο αριστερό τμήμα απεικονίζονται οι συμβεβλημένες τοποθεσίες της πλατφόρμας Organicity, στο μεσαίο τμήμα απεικονίζεται η πλατφόρμα Organicity με τις υποστηριζόμενες υπηρεσίες και τις διαθέσιμες διεπαφές APIs, ενώ στο δεξιό τμήμα υπάρχει η εφαρμογή πειραματισμού και οι διεπαφές χρήστη.

Οι τοποθεσίες της πλατφόρμας Organicity διαθέτουν τα *ενεργητικά τους στοιχεία* (assets), όπως συσκευές και πλατφόρμες ανοικτών δεδομένων, τα οποία μια τοποθεσία στοχεύει να διαμοιραστεί με τις υπηρεσίες της πλατφόρμας Organicity. Τα ενεργητικά στοιχεία μπορεί να είναι από την φύση τους ετερογενή. Για να είναι αυτά τα στοιχεία προσβάσιμα μέσα από τις υπηρεσίες της πλατφόρμας Organicity, ως μέρος του Αστικού Παρατηρητηρίου Δεδομένων (Urban Data Observatory), θα πρέπει να μπορούν να ανακαλυφθούν μέσω ενός τοπικού ευρετηρίου ενεργητικών στοιχείων και να διατεθούν μέσω των συμβεβλημένων διεπαφών εφαρμογών. Οι τοποθεσίες της πλατφόρμας Organicity πρέπει να πιστοποιηθούν και να εξουσιοδοτηθούν για να διαμοιράσουν τα ενεργητικά τους στοιχεία στις υπηρεσίες της πλατφόρμας Organicity, και η πλατφόρμα Organicity να επικυρώσει τις περιγραφές των εισερχόμενων ενεργητικών στοιχείων. Οι υπηρεσίες σε επίπεδο τοποθεσίας Organicity μπορούν να φιλοξενηθούν σε υποδομές τοπικών τοποθεσιών έχοντας πιθανές εναλλακτικές υλοποιήσεις των συμβεβλημένων διεπαφών εφαρμογών.

Τη στιγμή που μία νέα τοποθεσία επιθυμεί να συμμετάσχει στις υπηρεσίες της πλατφόρμας Organicity, ο εκπρόσωπος της τοποθεσίας, που μελλοντικά θα παίζει και το ρόλο του διαχειριστή, θα πρέπει να έρθει σε επικοινωνία με τους διαχειριστές της πλατφόρμας, ώστε να προχωρήσει η διαδικασία εγγραφής της νέας τοποθεσίας. Όταν πραγματοποιηθεί αυτή η διαδικασία και γίνει δεκτή η νέα τοποθεσία, οι υπηρεσίες της πλατφόρμας Organicity επιστρέφουν στην πρόσφατα συμβεβλημένη τοποθεσία ένα μοναδικό αναγνωριστικό και δίνουν τα απαραίτητα δικαιώματα πρόσβασης στον εν λόγω διαχειριστή της τοποθεσίας.

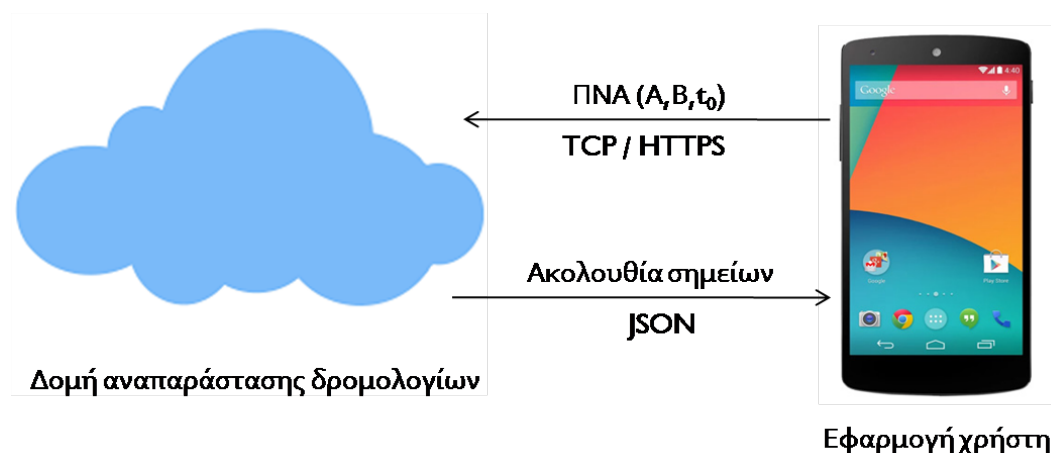
Ο διαχειριστής της τοποθεσίας είναι υπεύθυνος για τη διεκπεραίωση της διαδικασίας διαμόρφωσης που σχετίζεται με την νέα τοποθεσία της πλατφόρμας Organicity, καθορίζει τα ενεργητικά στοιχεία που εκείνη διαθέτει και κατηγοριοποιεί τις υπηρεσίες που συμμετέχουν στην πλατφόρμα πειραματισμού. Επιπλέον, προκειμένου να συμμορφωθούν με την πολιτική πιστοποίησης και εξουσιοδότησης, δημιουργούνται διακριτικά για κάθε πάροχο υπηρεσιών που

προσφέρει τα δεδομένα του μέσω μιας συμβεβλημένης τοποθεσίας, και οι διάφοροι διαχειριστές τοποθεσιών καθίστανται υπεύθυνοι για τη διαχείριση της διαμόρφωσης ώστε να παρέχουν σε τρίτους και στους χρήστες πρόσβαση στα διάφορα ενεργητικά στοιχεία που παρέχει η τοποθεσία τους.

Κάθε τοποθεσία της πλατφόρμας Organicity δίνει πρόσβαση σε μια σειρά ενεργητικών στοιχείων μέσω των τοπικών της καταλόγων, προκειμένου να τα διαθέσει στους πειραματιστές μέσω του κεντρικού καταλόγου ενεργητικών στοιχείων. Αυτά τα ενεργητικά στοιχεία μπορούν να παρέχονται από τις ίδιες τις τοποθεσίες, από τους τεχνολογικούς παρόχους που ενδιαφέρονται να συμμετάσχουν στις υπηρεσίες της πλατφόρμας Organicity, ή ακόμα και από τους πολίτες που στοχεύουν να συμμετάσχουν στο συν-δημιουργικό περιβάλλον της πλατφόρμας Organicity.

6.3 Ο Ταξιδιωτικός Πλοηγός

Ο ταξιδιωτικός πλοηγός είναι μια εφαρμογή βασισμένη σε υπολογιστικό νέφος. Η αρχιτεκτονική του υπολογιστικού νέφους περιγράφεται στην εργασία [44]. Το μέρος του εξυπηρετητή, που περιέχει την μηχανή δρομολόγησης, βρίσκεται στη δομή του υπολογιστικού νέφους, ενώ η εφαρμογή χρήστη βρίσκεται στην κινητή συσκευή.



Σχήμα 6.3: Ο ταξιδιωτικός πλοηγός.

Ο εξυπηρετητής ανταποκρίνεται στα αιτήματα των χρηστών για εύρεση βέλ-

τιστων διαδρομών μέσω των πρωτοκόλλων επικοινωνίας TCP ή HTTP/HTTPS. Τα αιτήματα εύρεσης βέλτιστης διαδρομής περιέχουν τα επιλεγμένα από τον χρήστη γεωγραφικά σημεία αφετηρίας και προορισμού, τον χρόνο αναχώρησης, και το επιθυμητό μέσο μεταφοράς. Για κάθε τέτοιο αίτημα, η υπηρεσία υπολογίζει τη ζητούμενη καλύτερη διαδρομή για το αντίστοιχο ερώτημα και τη στέλνει πίσω στον χρήστη. Το Σχήμα 6.3 παρουσιάζει ένα διάγραμμα λειτουργίας του ταξιδιωτικού οδηγού για απάντηση στο θεμελιώδες πρόβλημα που μελετήθηκε, το πρόβλημα νωρίτερης άφιξης (PNA) από σταθμό A σε σταθμό B με ώρα αναχώρησης μεταγενέστερη της t_0 .

Η έξοδος της υπηρεσίας νέφους είναι μια ακολουθία σημείων σε μορφή JSON, μαζί με πληροφορίες σχετικά με τα επιτρεπόμενα μέσα μεταφοράς και τους χρόνους αναχώρησης/άφιξης από/σε κάθε σημείο του ταξιδιού. Η υπηρεσία για τον υπολογισμό της διαδρομής χρησιμοποιεί σύνολα δεδομένων πίνακα δρομολογίων, στη μορφή General Transit Feed Specification (GTFS) [42], η οποία περιέχει στάσεις, σταθμούς, διάφορα μέσα δημόσιας συγκοινωνίας (π.χ. τρένα, λεωφορεία, κλπ), σταθμούς μετεπιβίβασης, και ώρες αναχώρησης/άφιξης των μέσων μαζικής μεταφοράς σε κάθε σταθμό/στάση. Τα δεδομένα πινάκων δρομολογίων αναπαριστώνται σαν ένα γράφημα του μοντέλου DTM.

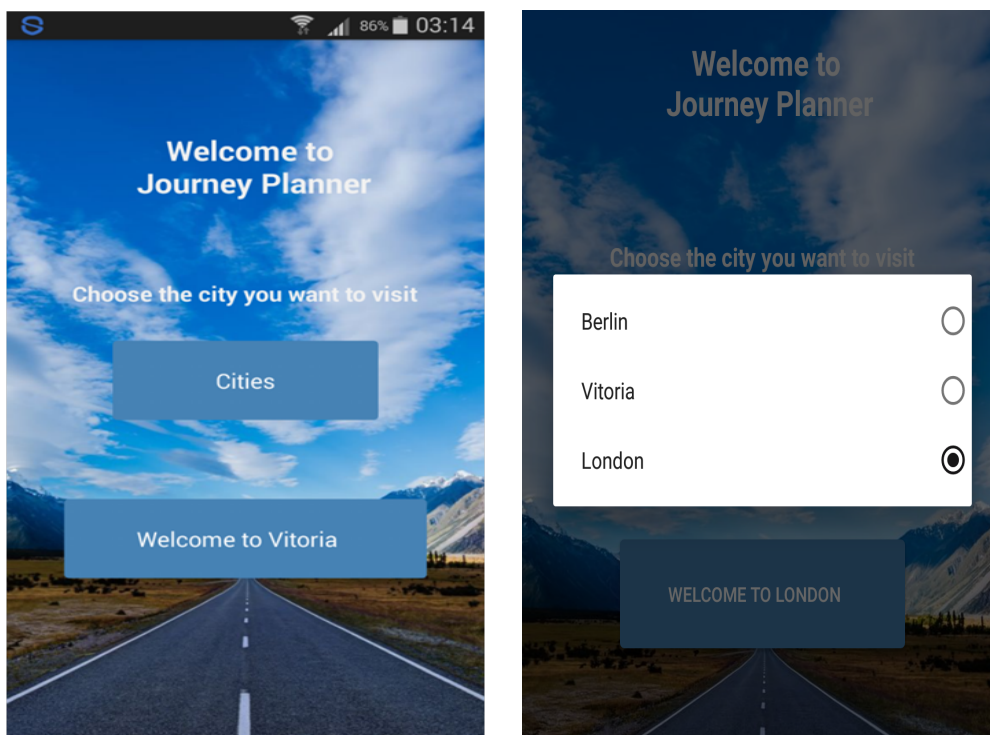
Η εφαρμογή κινητών συσκευών λειτουργεί σε έξυπνες συσκευές που υποστηρίζουν λειτουργικό σύστημα Android. Η διεπαφή χρήστη (UI) αναπτύχθηκε σε περιβάλλον Android Studio. Λεπτομέρειες υλοποίησης της διεπαφής χρήστη περιγράφονται στην εργασία [82]. Αναπτύχθηκαν ακόμη οι επικοινωνίες μεταξύ διεπαφής χρήστη και υπολογιστικού νέφους, καθώς και προς/από άλλες υπηρεσίες, συμπεριλαμβανομένης και αυτής της αξιολόγησης της προτεινόμενης από τον ταξιδιωτικό οδηγό διαδρομής.

6.3.1 Λειτουργικότητα Εφαρμογής

Η λειτουργία της εφαρμογής κινητών συσκευών αποτυπώνεται στα Σχήματα 6.4, 6.5, 6.6, 6.7, και 6.8.

Η αρχική εικόνα της εφαρμογής αποτυπώνεται στο Σχήμα 6.4. Ο χρήστης με το πάτημα της επολογής «Cities» έχει την δυνατότητα να επιλέξει την πόλη

που επιθυμεί να πλοηγηθεί. Στο ίδιο σχήμα απεικονίζεται και το βήμα της

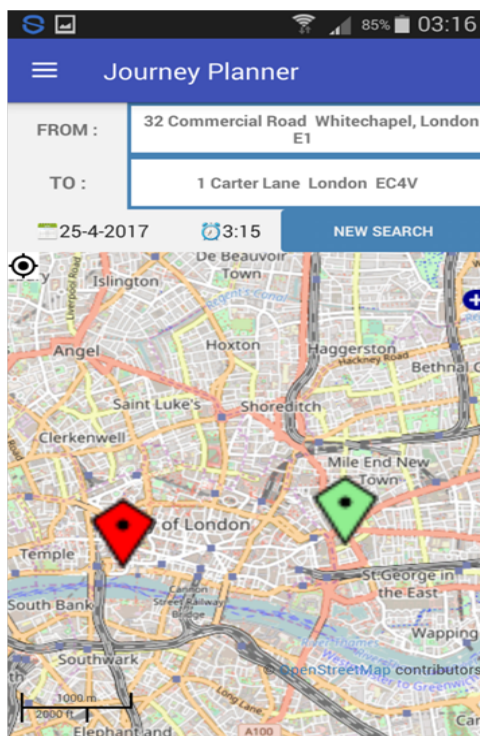


Σχήμα 6.4: Αριστερά: αρχική οθόνη ταξιδιωτικού πλοηγού. Δεξιά: επιλογή πόλης.

επιλογής πόλης, όπου εμφανίζονται στον χρήστη η λίστα με τις διαθέσιμες πόλεις στις οποίες μπορεί να προγραμματίσει το ταξίδι του.

Στο Σχήμα 6.5 απεικονίζεται το βήμα του πλοηγού στο οποίο ο χρήστης δίνει αφετηρία, προορισμό, και τον επιθυμητό χρόνο αναχώρησης για την εκτέλεση ενός ερωτήματος νωρίτερης άφιξης (NA). Αρχικά εμφανίζεται στον χρήστη ο χάρτης της πόλης που έχει επιλέξει. Υπάρχουν δύο σημάδια επάνω στο χάρτη τα οποία μπορεί με ευκολία να μετακινήσει ο χρήστης, το πράσινο σημάδι προσδιορίζει το σημείο αφετηρίας και το κόκκινο το σημείο τερματισμού της ζητούμενης διαδρομής. Επιπλέον, υπάρχει η δυνατότητα να εστιάσει (zoom in, zoom out) στα επιθυμητά σημεία ώστε να τα προσδιοριστούν με μεγαλύτερη ακρίβεια. Στο μέρος πάνω από το χάρτη, φαίνεται η ημερομηνία και η ώρα, τα οποία μπορεί να διαχειριστεί ο χρήστης προκειμένου να προσδιορίσει την ημερομηνία και την χρονική στιγμή εκκίνησης από την αφετηρία. Αφού

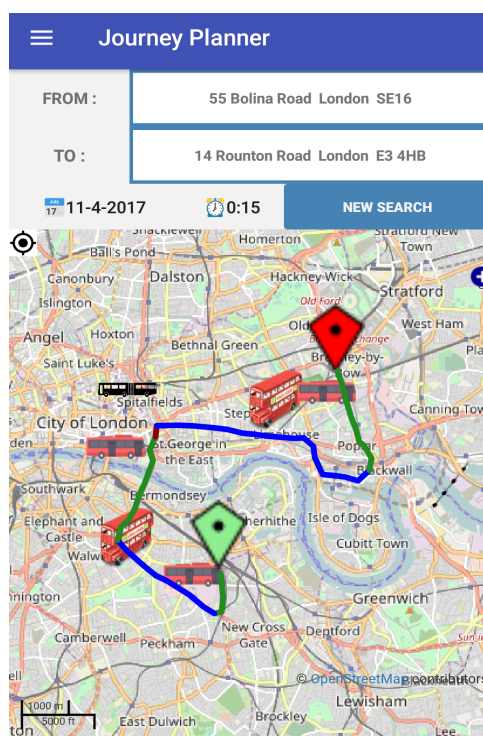
οριστούν τα παραπάνω, ο χρήστης πατά την επιλογή «NEW SEARCH» για τον υπολογισμό και την εμφάνιση της βέλτιστης διαδρομής.



Σχήμα 6.5: Είσοδος στοιχείων για ερώτημα εύρεσης βέλτιστης διαδρομής νωρίτερης άφιξης.

Η βέλτιστη διαδρομή που υπολογίζεται, αποστέλλεται από τον εξυπηρετητή που βρίσκεται στην υποδομή υπολογιστικού νέφους και προβάλεται πάνω στον χάρτη. Στο Σχήμα 6.6 απεικονίζεται το αποτέλεσμα που προκύπτει από την εκτέλεση ενός ερωτήματος νωρίτερης άφιξης (NA) και δίνεται η χρονική στιγμή άφιξης στον τερματικό σταθμό.

Στο Σχήμα 6.7 απεικονίζονται οι οδηγίες πλοήγησης, προκειμένου ο χρήστης να γνωρίζει ποια MMM πρέπει να ακολουθήσει και σε ποιους σταθμούς θα μετεπιβιβασθεί. Η ακρίβεια των οδηγιών εξαρτάται από την ποιότητα των παρεχόμενων δεδομένων και όχι από την εφαρμογή του ταξιδιωτικού οδηγού. Έτσι, δίνονται αριθμημένα οι πληροφορίες για τον χρήστη που είναι το πλήθος και το είδος των οχημάτων που θα χρησιμοποιήσει, το χρόνο που θα δαπανήσει σε κάθε μεταφορικό μέσο, και πιθανές στάσεις που απαιτούνται για την αλλαγή

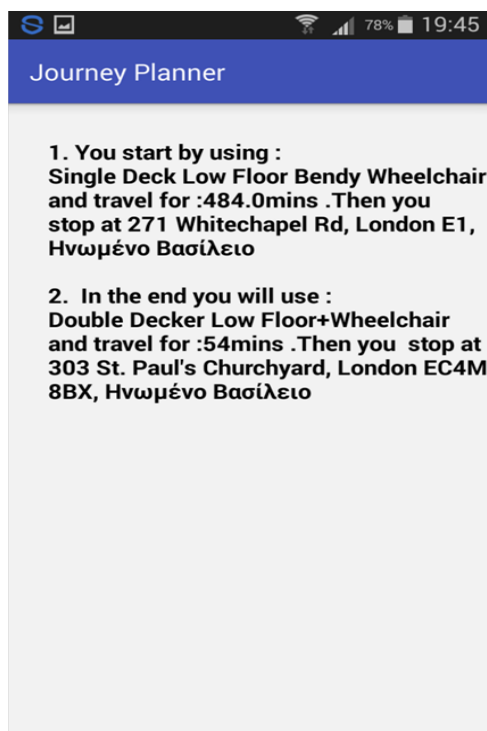


Σχήμα 6.6: Ερώτηση και απεικόνιση βέλτιστης διαδρομής.

του μέσου μεταφοράς.

Το Σχήμα 6.8 απεικονίζεται η δυνατότητα επιλογής μέσου μαζικής μεταφοράς που έχει ο χρήστης σε ένα ερώτημα ΝΑ. Στην πάνω αριστερή γωνία της οθόνης υπάρχουν τρεις οριζόντιες γραμμές, τις οποίες αν επιλέξει ο χρήστης, θα εμφανιστούν επιπλέον επιλογές. Έτσι υπάρχει η δυνατότητα επιλογής από τον ίδιο τον χρήστη του μεταφορικού μέσου που θα χρησιμοποιήσει σύμφωνα με τις ανάγκες και τις προτιμήσεις του. Οι διαθέσιμες επιλογές είναι μετακίνηση με τα πόδια, το αυτοκίνητο, το λεωφορείο, το τρένο και το μετρό, εφόσον η πόλη που γίνεται η αναζήτηση το περιέχει. Στην κορυφή της οθόνης υπάρχει η επιλογή «Combined Route» που δίνει τη δυνατότητα συνδυασμού των παραπάνω τρόπων μεταφοράς. Τέλος, ακολουθεί ένα σύνολο χρήσιμων πληροφοριών για το χρόνο, την απόσταση και την εξοικονόμηση ρύπων που θα εξασφαλίσει ακολουθώντας την προτεινόμενη βέλτιστη διαδρομή.

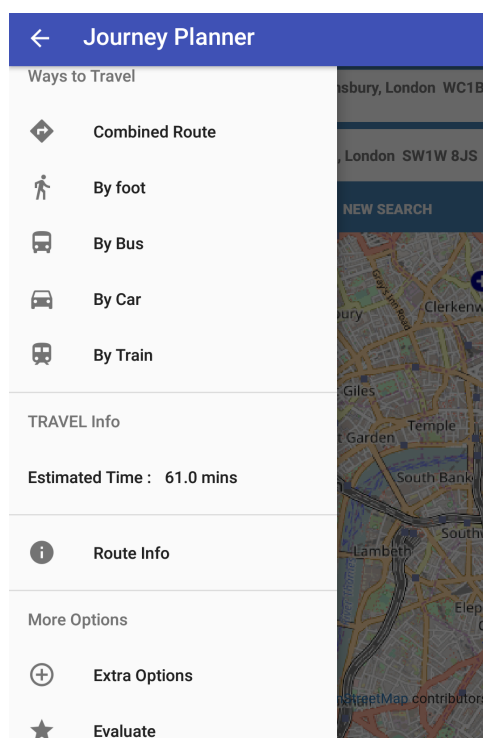
Η πρακτική εφαρμογή του ταξιδιωτικού πλοηγού αξιολογήθηκε σε πιλοτική εφαρμογή που εκτελέστηκε στη δομή υπολογιστικού νέφους του έργου [65]



Σχήμα 6.7: Οδηγίες πλοήγησης.

στην πόλη της Vitoria-Gasteiz το φθινόπωρο 2016 και περιγράφεται στην εργασία [44]. Το δημόσιο δίκτυο συγκοινωνιών της Vitoria-Gasteiz αποτελείται από 355 σταθμούς και από 129,050 στοιχειώδεις συνδέσεις, με αποτέλεσμα να δημιουργείται ένα γράφημα DTM περίπου $113K$ κορυφών και περίπου $335K$ ακμών. Ο ταξιδιωτικός πλοηγός πέτυχε κατά μέσο όρο χρόνο $0.15ms$ για απάντηση ερωτημάτων και χρόνο $1.2\mu s$ για ενημέρωση του πίνακα δρομολογίων μετά από μια καθυστέρηση.

Υπενθυμίζεται ότι στο Κεφάλαιο 5 διεξήχθη μια εκτενής πειραματική μελέτη σε πραγματικούς πίνακες δρομολογίων [49, 80, 81]. Αυτά τα σύνολα δεδομένων περιείχαν (μεταξύ άλλων) τα μητροπολιτικά δίκτυα δημόσιων συγκοινωνιών της Αθήνας, του Βερολίνου, του Λονδίνου και της Μαδρίτης. Το μεγαλύτερο σε μέγεθος μητροπολιτικό δίκτυο ήταν του Λονδίνου, αποτελούμενο από 20,843 σταθμούς και 14,064,967 συνδέσεις, με αποτέλεσμα να προκύπτει ένα γράφημα DTM με 14,085,810 κορυφές και 41,837,355 ακμές. Σε αυτή την περίπτωση, η απόδοση του ταξιδιωτικού πλοηγού στην ίδια υποδομή υπολογιστικού νέφους



Σχήμα 6.8: Επιλογή μέσου μεταφοράς.

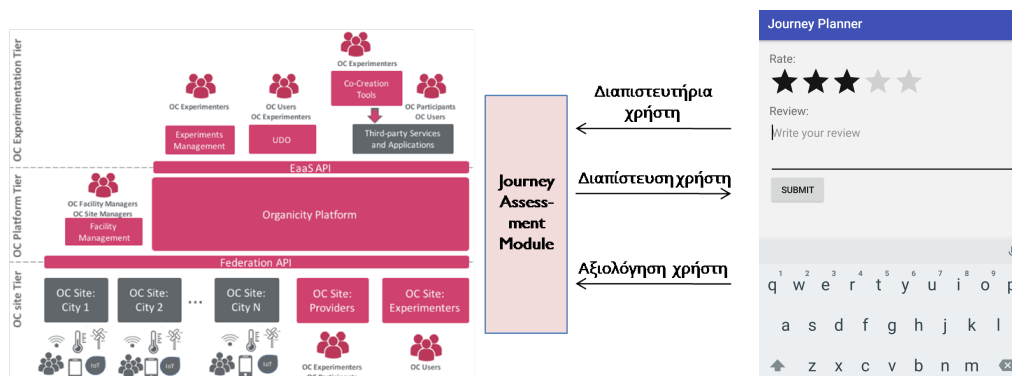
[65] ήταν αμελητέα επιβαρυμένη σε σχέση με τα αποτελέσματα του Κεφαλαίου 5 (μέσος χρόνος απόκρισης ερωτημάτων 10.25ms και μέσος χρόνος ενημέρωσης πίνακα δρομολογίων μετά από μια καθυστέρηση 271.46μs).

6.3.2 Υπηρεσία Αξιολόγησης Διαδρομών

Προκειμένου να βοηθηθούν οι χρήστες στην αξιολόγηση των παρεχόμενων διαδρομών από τον ταξιδιωτικό πλοηγό, αναπτύχθηκε η *υπηρεσία αξιολόγησης διαδρομών* (journey assessment service – JAS).

Η υπηρεσία αξιολόγησης διαδρομών αναπτύχθηκε χρησιμοποιώντας τις βασικές υπηρεσίες που προσφέρει το πλαίσιο Πειραματισμού-ως-Υπηρεσία (Experimentation-as-a-Service) της πλατφόρμας Organicity [71]. Η διασύνδεση της με την πλατφόρμα Organicity φαίνεται στο Σχήμα 6.9.

Η υπηρεσία λαμβάνει τις αξιολογήσεις των διαδρομών που πραγματοποιήσαν οι χρήστες του ταξιδιωτικού πλοηγού με τις κινητές συσκευές τους. Η



Σχήμα 6.9: Η υπηρεσία αξιολόγησης διαδρομών.

επικοινωνία γίνεται με αιτήματα τύπου HTTPS POST σε τρία βήματα:

- Ο χρήστης στέλνει τα διαπιστευτήριά του για εξουσιοδότηση.
- Η υπηρεσία αξιολόγησης διαδρομών δίνει πρόσβαση στον χρήστη εξουσιοδοτώντας τον να στείλει την αξιολόγησή του.
- Ο χρήστης στέλνει την αξιολόγηση της διαδρομής.

Τα δεδομένα της αξιολόγησης της διαδρομής, που μεταδίδονται σε μορφή JSON, αποτελούνται από την βαθμολογία της αξιολόγησης του χρήστη και την κριτική του σχετικά με την διαδρομή που του προτάθηκε.

Η επιλογή για την ενεργοποίηση της αξιολόγησης της διαδρομής (Σχήμα 6.8, επιλογή «Evaluate») ενεργοποιεί την αξιολόγηση όπως αποτυπώνεται στο Σχήμα 6.10, όπου οι χρήστες κινητών συσκευών μπορούν να υποβάλουν την βαθμολογία τους και την κριτική τους.

Journey Planner

Rate:

★★★★☆

Review:

Write your review

SUBMIT

1 2 3 4 5 6 7 8 9 0

q w e r t y u i o p

a s d f g h j k l

↑ z x c v b n m ✕

Σχήμα 6.10: Αξιολόγηση διαδρομής.

Κεφάλαιο 7

Συμπεράσματα και Προοπτικές

Στο κεφάλαιο αυτό παρουσιάζονται τα βασικά συμπεράσματα της διδακτορικής διατριβής και συζητώνται μελλοντικές επεκτάσεις και προοπτικές της.

7.1 Συμπεράσματα

Στην παρούσα διδακτορική διατριβή μελετήθηκε το πρόβλημα της ταξιδιωτικής πλοήγησης και γενικότερα των υπηρεσιών κινητικότητας σε σύγχρονα αστικά περιβάλλοντα με χρήση MMM, και τα θεμελιώδη αλγοριθμικά προβλήματα κινητικότητας πολιτών που προκύπτουν. Η χρήση MMM συνεπάγεται την ανάγκη αποδοτικής αναπαράστασης και διαχείρισης πινάκων δρομολογίων που περιγράφουν τα δρομολόγια των MMM.

Η έρευνα επικεντρώθηκε στα ρεαλιστικά γραφοκεντρικά μοντέλα αναπαράστασης πινάκων δρομολογίων και τη διαχείρισή τους σε δυναμικά σενάρια, δηλαδή σενάρια κατά τα οποία ο πίνακας δρομολογίων μπορεί να αλλάζει (απρόβλεπτα) όταν λαμβάνουν χώρα καθυστερήσεις των MMM. Σε αυτό το πλαίσιο, παρουσιάστηκαν δύο νέα γραφοκεντρικά μοντέλα, το ανηγμένο χρονοεκτεταμένο μοντέλο και το δυναμικό μοντέλο πινάκων δρομολογίων DTM, μαζί με αποδοτικούς αλγορίθμους απάντησης ερωτημάτων εύρεσης βέλτιστων διαδρομών και ενημέρωσης της πληροφορίας των πινάκων δρομολογίων σε περίπτωση καθυστερήσεων, οι οποίοι αξιολογήθηκαν πειραματικά σε πραγματικά δεδομένα. Παρουσιάστηκε επίσης ένας ψηφιακός ταξιδιωτικός πλοηγός για έ-

ξυπνες κινητές συσκευές, βασισμένος σε δομή υπολογιστικού νέφους, ο οποίος επιτρέπει και την αξιολόγηση των προτεινόμενων διαδρομών από τους χρήστες.

Τα σημαντικότερα συμπεράσματα που προέκυψαν από την παρούσα διδακτορική διατριβή είναι τα εξής:

- Τα δύο νέα γραφοκεντρικά μοντέλα δίνουν βελτιωμένες χωρο-αποδοτικές και εύκολα διαχειρίσιμες αναπαραστάσεις πινάκων δρομολογίων.
- Οι νέοι αλγόριθμοι απάντησης ερωτημάτων εύρεσης βέλτιστων διαδρομών και ενημέρωσης της πληροφορίας των πινάκων δρομολογίων σε περίπτωση καθυστερήσεων είναι αποδοτικότεροι σε σχέση με παλαιότερες μεθόδους γραφοκεντρικών μοντέλων.
- Η ενίσχυση των αλγορίθμων απάντησης ερωτημάτων με ευρετικές τεχνικές, και ειδικά με την ενσωμάτωση της ευρετικής τεχνικής ALT, βελτίωσε σημαντικά την απόδοση τους.
- Η εκτενής πειραματική αξιολόγηση σε μια πληθώρα πραγματικών πινάκων δρομολογίων κατέδειξε ότι το μοντέλο DTM είναι αποδοτικότερο στην ενημέρωση του πίνακα δρομολογίων σε περίπτωση καθυστερήσεων, ενώ το ανηγμένο χρονοεκτεταμένο μοντέλο είναι καλύτερο στο χρόνο απόκρισης ερωτημάτων. Η πειραματική αξιολόγηση κατέδειξε επίσης ότι τα γραφοκεντρικά μοντέλα μπορούν να επιτύχουν χρόνους απόκρισης ερωτημάτων συγκρίσιμους με (ή και καλύτερους από) εκείνους των διανυσματοκεντρικών μοντέλων.
- Η πειραματική αξιολόγηση σε πραγματικά δεδομένα, σε συνδυασμό με την ανάπτυξη της εφαρμογής κινητών συσκευών, κατέδειξαν την πρακτική εφαρμοσιμότητα των προτεινόμενων γραφοκεντρικών μοντέλων και αλγορίθμων.

7.2 Προοπτικές

Οι επεκτάσεις και οι γενικότερες προοπτικές της παρούσας διδακτορικής διατριβής έχουν ως εξής.

Τα γραφοκεντρικά μοντέλα μπορούν να επεκταθούν λαμβάνοντας υπόψη:

- (α) Το διαχωρισμό των διαφόρων τύπων MMM, έτσι ώστε να απαντώνται ερωτήματα *πολυτροπικών* βέλτιστων διαδρομών (multimodal route planning).
- (β) Τις διαφορετικές πολιτικές διαχείρισης καθυστερήσεων.
- (γ) Τους χρόνους ανθεκτικότητας (slack times) των δρομολογίων στους σταθμούς.
- (δ) Την ύπαρξη πεζόδρομων μεταξύ των σταθμών. Η ενσωμάτωση των πεζόδρομων μπορεί εύκολα να ενταχθεί στα γραφοκεντρικά μοντέλα με αποδοτικό τρόπο. Στα χρονοεκτεταμένα μοντέλα, αν υπάρχει ένας πεζόδρομος από έναν σταθμό (ή μια στάση ενός σταθμού) p_i σε έναν άλλο p_j , τότε για κάθε γεγονός άφιξης στον σταθμό p_i προστίθεται μία ακμή στην πρώτη προσβάσιμη κορυφή μετεπιβίβασης του σταθμού p_j . Στο μοντέλο DTM, αρκεί μια μικρή τροποποίηση του αλγορίθμου απάντησης ερωτημάτων ως ακολούθως: όταν η κορυφή μεταγωγής ενός συγκεκριμένου σταθμού p_i διαγράφεται από την ουρά προτεραιότητας, τότε, στην περίπτωση που υπάρχει πεζόδρομος προς έναν γειτονικό σταθμό p_j , ο χρόνος άφιξης στον p_j μέσω μιας κορυφής αναχώρησης πρέπει απλά να συγκριθεί με το χρόνο άφιξης στον p_j μέσω του πεζόδρομου, ο οποίος δίνεται από το άθροισμα του χρόνου άφιξης στον p_i συν το κόστος της διαδρομής του πεζόδρομου mod 1440.

Άλλες προοπτικές μελλοντικής έρευνας περιλαμβάνουν τις εξής περιπτώσεις. Μία προοπτική είναι ο συνδυασμός των δύο νέων γραφοκεντρικών μοντέλων με άλλες γνωστές τεχνικές επιτάχυνσης. Σε αυτή την κατεύθυνση, οι πιο ελπιδοφόρες είναι αυτές που βασίζονται στην τεχνική Arc-flags [60], καθώς μπορούν να συνδυαστούν με την ευρετική τεχνική ALT [27] και να υποστηρίξουν δυναμικές ενημερώσεις [21]. Επιπλέον, η αποδοτική υλοποίηση της ευρετικής τεχνικής ALT που παρουσιάζεται στην εργασία [37] θα μπορούσε να βελτιώσει περαιτέρω τον χρόνο απόκρισης ερωτημάτων. Δεδομένων αυτών των συνδυασμών, θα ήταν επίσης ενδιαφέρον να συγκριθούν πειραματικά τα δύο

νέα γραφοκεντρικά μοντέλα με άλλα διανυσματοκεντρικά μοντέλα και χρονο-εξαρτώμενες προσεγγίσεις.

Μια άλλη προοπτική είναι η αντιμετώπιση πολυκριτηριακών προβλημάτων και η αντίστοιχη επέκταση της εφαρμογής του ταξιδιωτικού πλοηγού προς αυτή την κατεύθυνση.

Βιβλιογραφία

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, 1993.
- [2] T. Akiba, Y. Iwata, and Y. Yoshida. Dynamic and historical shortest-path distance queries on large evolving networks by pruned landmark labeling. In *World Wide Web - WWW 2014*, ACM, pp. 237-248.
- [3] Android. <https://www.android.com/>.
- [4] Android Studio. <http://developer.android.com/index.html>.
- [5] H. Bast, E. Carlsson, A. Eigenwillig, R. Geisberger, C. Harrelson, V. Raychev, and F. Viger. Fast routing in very large public transportation networks using transfer patterns. In *Algorithms - ESA 2010, Lecture Notes in Computer Science*, Vol. 6346 (Springer 2010), pp. 290-301.
- [6] H. Bast, D. Delling, A. V. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. F. Werneck. Route planning in transportation networks. In *Algorithm Engineering - Selected Results and Surveys, Lecture Notes in Computer Science*, Vol. 9220 (Springer 2016), pp. 19-80.
- [7] H. Bast, J. Sternisko, and S. Storandt. Delay-Robustness of Transfer Patterns in Public Transportation Route Planning. In *Algorithmic Approaches for Transportation Modeling, Optimization, and Systems - ATMOS 2013, OASICS Vol. 33* (2013), pp. 42-54.

- [8] R. Bauer, D. Delling, P. Sanders, D. Schieferdecker, D. Schultes, and D. Wagner. Combining hierarchical and goal-directed speed-up techniques for Dijkstra’s algorithm. *ACM Journal of Experimental Algorithmics*, 15:2.3, 2010.
- [9] R. Bauer, D. Delling, and D. Wagner. Experimental study of speed up techniques for timetable information systems. *Networks*, 57(1):38–52, 2011.
- [10] N. Baumann, R. Schmidt. Buxtehude-Garmisch in 6 Sekunden: Die elektronische Fahrplanauskunft (EFA) der Deutschen Bundesbahn. *Die Bundesbahn - Zeitschrift für aktuelle Verkehrsfragen*, 10:929–931, 1988.
- [11] N.R. Brisaboa, D. Caro, A. Fariña, and M.A. Rodríguez. A compressed suffix-array strategy for temporal-graph indexing. In *String Processing and Information Retrieval – SPIRE 2014*, pp. 77–88.
- [12] N.R. Brisaboa, A. Fariña, G. Navarro, and J.R. Paramá. Lightweight natural language text compression. *Information Retrieval*, 10(1):1–33, 2007.
- [13] G. Brodal. Worst-Case Efficient Priority Queues. In *ACM-SIAM Symposium on Discrete Algorithms – SODA 1996*, pp. 52–58.
- [14] F. Bruera, S. Cicerone, G. D’Angelo, G. D. Stefano, and D. Frigioni. Dynamic multi-level overlay graphs for shortest paths. *Mathematics in Computer Science*, 1(4):709–736, 2008.
- [15] D. Caro, M.A. Rodríguez, and N.R. Brisaboa. Data structures for temporal graphs based on compact sequence representations. *Information Systems*, 51:1–26, 2015.
- [16] S. Cicerone, G. D’Angelo, G. D. Stefano, D. Frigioni, and A. Navarra. Delay management problem: Complexity results and robust algorithms. In *Combinatorial Optimization and Applications – COCOA 2008, Lecture Notes in Computer Science*, Vol. 5165 (Springer 2008), pp. 458–468.

- [17] S. Cicerone, G. D'Angelo, G. Di Stefano, D. Frigioni, A. Navarra, M. Schachtebeck, and A. Schöbel. Recoverable robustness in shunting and timetabling. In *Robust and Online Large-Scale Optimization, Lecture Notes in Computer Science*, Vol. 5868 (Springer 2009), pp. 28–60.
- [18] S. Cicerone, G. D'Angelo, G. Di Stefano, D. Frigioni, and A. Navarra. Recoverable robust timetabling for single delay: Complexity and polynomial algorithms for special cases. *Journal of Combinatorial Optimization*, 18(3):229–257, 2009.
- [19] A. Cionini, G. D'Angelo, M. D'Emidio, D. Frigioni, K. Giannakopoulou, A. Paraskevopoulos, and C. D. Zaroliagis. Engineering graph-based models for dynamic timetable information systems. In *Algorithmic Approaches for Transportation Modelling, Optimization, and Systems – ATMOS 2014, OASICS Vol. 42* (2014), pp. 46–61.
- [20] A. Cionini, G. D'Angelo, M. D'Emidio, D. Frigioni, K. Giannakopoulou, A. Paraskevopoulos, and C. D. Zaroliagis. Engineering graph-based models for dynamic timetable information systems. *Journal of Discrete Algorithms*, to appear.
- [21] G. D'Angelo, M. D'Emidio, and D. Frigioni. Fully dynamic update of arc-flags. *Networks*, 63(3):243–259, 2014.
- [22] G. D'Angelo, M. D'Emidio, D. Frigioni, and C. Vitale. Fully dynamic maintenance of arc-flags in road networks. In *Experimental Algorithms – SEA 2012, Lecture Notes in Computer Science*, Vol. 7276 (Springer 2012), pp. 135–147.
- [23] E.W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [24] S. E. Dreyfus. An appraisal of some shortest-path algorithms. *Operations Research*, 17(3):395–412, 1969.

- [25] D. Delling, J. Dibbelt, T. Pajor, and R. Werneck. Public transit labeling. In *Experimental Algorithms – SEA 2015, Lecture Notes in Computer Science*, Vol. 9125 (Springer 2015), pp. 273–285.
- [26] D. Delling, K. Giannakopoulou, D. Wagner, and C. Zaroliagis. Timetable Information Updating in Case of Delays: Modeling Issues. Technical Report ARRIVAL-TR-0133, ARRIVAL Project, 2008.
- [27] D. Delling, T. Pajor, and D. Wagner. Engineering time-expanded graphs for faster timetable information. In *Robust and Online Large-Scale Optimization, Lecture Notes in Computer Science* Vol. 5868 (Springer 2009), pp. 182–206.
- [28] D. Delling, T. Pajor, and R. F. Werneck. Round-based public transit routing. In *14th Workshop on Algorithm Engineering and Experiments – ALENEX 2012, SIAM*, pp. 130–140.
- [29] D. Delling, T. Pajor, and R. F. Werneck. Round-based public transit routing. *Transportation Science*, 49(3):591–604, 2015.
- [30] D. Delling and D. Wagner. Landmark-based routing in dynamic graphs. In *Experimental Algorithms – WEA 2007, Lecture Notes in Computer Science*, Vol. 4525 (Springer 2007), pp. 52–65.
- [31] D. Delling and R. F. Werneck. Faster customization of road networks. In *Symposium on Experimental Algorithms – SEA 2013, Lecture Notes in Computer Science*, Vol. 7933 (Springer 2013), pp. 30–42.
- [32] J. Dibbelt, D. Gavalas, K. Giannakopoulou, D. Kehagias, C. Konstantopoulos, S. Kontogiannis, G. Pantziou, D. Wagner, and C. Zaroliagis. Eco-aware Vehicle Routing in Urban Environments. In *Proc. 22nd IEEE Symposium on Computers and Communications – ISCC 2017*, to appear.
- [33] J. Dibbelt, T. Pajor, B. Strasser, and D. Wagner. Intriguingly simple and fast transit routing. In *Experimental Algorithms – SEA 2013, Lecture Notes in Computer Science*, Vol. 7933 (Springer 2013), pp. 43–54.

- [34] J. Dibbelt, B. Strasser, and D. Wagner. Customizable contraction hierarchies. In *Experimental Algorithms – SEA 2014, Lecture Notes in Computer Science*, Vol. 8504 (Springer 2014), pp. 271–282.
- [35] J. Doran. An Approach to Automatic Problem-Solving. *Machine Intelligence*, 1:105–127, 1967.
- [36] EFA: Electronic Journey Planner. <http://www.mentz.net/en/verkehrsanskunft/>.
- [37] A. Efentakis and D. Pfoser. Optimizing Landmark-Based Routing and Preprocessing. In *6th ACM SIGSPATIAL Int. Work. on Computational Transp. Science – IWCTS 2013*, pp. 25:25–25:30.
- [38] D. Firmani, G. F. Italiano, L. Laura, and F. Santaroni. Is Timetabling Routing Always Reliable for Public Transport? In *Algorithmic Approaches for Transportation Modeling, Optimization, and Systems – ATMOS 2013, OASICS Vol. 33* (2013), pp. 15–26.
- [39] M. Fischetti, D. Salvagnin, and A. Zanette. Fast approaches to improve the robustness of a railway timetable. *Transportation Science*, 43(3):321–335, 2009.
- [40] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM* 34(3):596–615, 1987.
- [41] R. Geisberger, P. Sanders, D. Schultes, and D. Delling. Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In *Workshop Experimental Algorithms – WEA 2008, Lecture Notes in Computer Science*, Vol. 5038 (Springer 2008), pp. 319–333.
- [42] General Transit Feed Specification. <https://developers.google.com/transit/gtfs>.
- [43] K. Giannakopoulou. Optimal Routes in Dynamic Timetable Information Systems. MSc Thesis, Dept of Computer Engineering & Informatics, University of Patras, 2011.

- [44] K. Giannakopoulou, S. Kontogiannis, G. Papastavrou, and C. Zaroliagis. A Cloud-based Time-Dependent Routing Service. In *Algorithmic Aspects of Cloud Computing – ALGO CLOUD 2016, Lecture Notes in Computer Science*, Vol. 10230 (Springer 2017), pp. 41-64.
- [45] K. Giannakopoulou, S. Nikolettseas, A. Paraskevopoulos, and C. Zaroliagis. Dynamic Timetable Information in Smart Cities. In *Proc. 22nd IEEE Symposium on Computers and Communications – ISCC 2017*, to appear.
- [46] Giraph. <http://giraph.apache.org/>
- [47] A. Goldberg and C. Harrelson. Computing the shortest path: A* search meets graph theory. In *ACM-SIAM Symposium on Discrete Algorithms – SODA 2005*, (SIAM, 2005), pp. 156-165.
- [48] Google transit. <https://maps.google.com/landing/transit/index.html>
- [49] HaCon - Ingenieurgesellschaft mbH. <http://www.hacon.de>, 2008.
- [50] Timetable Information System HAFAS. <http://www.hacon.de/hafas-en/>.
- [51] P. E. Hart, N. J. Nilsson, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4:2, 1968.
- [52] W. Huo, and V.J. Tsotras. Efficient temporal shortest path queries on evolving social graphs. In *Scientific and Statistical Database Management – SSDBM 2014*, ACM, pp. 38:1-38:4.
- [53] D. B. Johnson. Priority queues with update and finding minimum spanning trees. *Information Processing Letters*, 4(3):53-57, 1975.
- [54] U. Khurana, and A. Deshpande, Efficient snapshot retrieval over historical graph data. In *IEEE International Conference on Data Engineering – ICDE 2013*, pp. 997-1008.

- [55] G. Koloniari, D. Souravlias, and E. Pitoura. On graph deltas for historical queries. In *Workshop on Online Social Systems – WOSS*, 2012.
- [56] A. Kosmatopoulos, K. Giannakopoulou, A. Papadopoulos, and K. Tsichlas. An Overview of Methods for Handling Evolving Graph Sequences. In *Algorithmic Aspects of Cloud Computing – ALGOCLOUD 2015, Lecture Notes in Computer Science*, Vol. 9511 (Springer 2016), pp. 181–192.
- [57] A. Kosmatopoulos, K. Tsichlas, A. Gounaris, S. Sioutas, and E. Pitoura. HiNode: an asymptotically space-optimal storage model for historical queries on graphs. *Distributed and Parallel Databases*, 35(4-5):249–285, 2017.
- [58] A.G. Labouseur, J. Birnbaum, P.W. Olsen, S.R. Spillane, J. Vijayan, J. Hwang, and W. Han. The G* graph database: efficiently managing large distributed dynamic graphs. *Distributed and Parallel Databases*, 33(4):479–514, 2015.
- [59] A.G. Labouseur, P.W. Olsen, and J. Hwang. Scalable and robust management of dynamic graph data. In *Proc. International Workshop on Big Dynamic Distributed Data – BD3 2013*, pp. 43–48.
- [60] U. Lauther. An Extremely Fast, Exact Algorithm for Finding Shortest Paths. In *Static Networks with Geographical Background*, 22:219–230, 2004.
- [61] J. Leskovec, and A. Krevl. SNAP – Stanford large network dataset collection. <http://snap.stanford.edu/data>, 2014.
- [62] C. Liebchen, M. Schachtebeck, A. Schöbel, S. Stiller, and A. Prigge. Computing delay resistant railway timetables. *Computers & OR*, 37(5):857–868, 2010.
- [63] G. Malewicz, M.H. Austern, , A.J.C. Bik, J.C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel: a system for large-scale graph processing. In *Proc. ACM International Conference on Management of Data – SIGMOD 2010*, pp. 135–146.

- [64] G. Mali, P. Michail, A. Paraskevopoulos, and C. Zaroliagis. A new dynamic graph structure for large-scale transportation networks. In *Algorithms and Complexity – CIAC 2013, Lecture Notes in Computer Science*, Vol. 7878 (Springer 2013), pp. 312–323.
- [65] MOVESMART project. <http://www.movesmartfp7.eu/>.
- [66] M. Müller-Hannemann and M. Schnee. Efficient timetable information in the presence of delays. In *Robust and Online Large-Scale Optimization, Lecture Notes in Computer Science*, Vol.5868 (Springer 2009), pp. 249–272.
- [67] Neo4j Graph Database. <http://neo4j.org/>
- [68] T.A.J. Nicholson. Finding the Shortest Route Between Two Points in a Network. *Computer Journal*, 9:275–280, 1966.
- [69] V. Nicosia, J. Tang, C. Mascolo, M. Musolesi, G. Russo, and V. Latora. Graph metrics for temporal networks. In *Temporal Networks* (Springer 2013), pp. 15–40.
- [70] Open HandSet Alliance. <http://www.openhandsetalliance.com/>.
- [71] ORGANICITY project. <http://organicity.eu/>.
- [72] E. Pyrga, F. Schulz, D. Wagner and C. Zaroliagis. Efficient Models for Timetable Information in Public Transportation Systems, *ACM Journal of Experimental Algorithmics*, 12(2.4):1–39, 2008.
- [73] C. Ren, E. Lo, B. Kao, X. Zhu and R. Cheng. On querying historical evolving graph sequences. *PVLDB* 4(11):726–737, 2011.
- [74] M. Schachtebeck and A. Schöbel. To wait or not to wait - and who goes first? Delay management with priority decisions. *Transportation Science*, 44(3):307–321, 2010.

- [75] D. Schultes and P. Sanders. Dynamic highway-node routing. In *Experimental Algorithms – WEA 2007, Lecture Notes in Computer Science*, Vol. 4525 (Springer 2007), pp. 66–79.
- [76] K. Semertzidis, E. Pitoura, and K. Lillis. Timereach: Historical reachability queries on evolving graphs. In *Extending Database Technology – EDBT 2015*, pp. 121–132.
- [77] B. Shao, H. Wang, and Y. Li. Trinity: a distributed graph engine on a memory cloud. In *Proc. ACM International Conference on Management of Data – SIGMOD 2013*, pp. 505–516.
- [78] S.R. Spillane, J. Birnbaum, D. Bokser, D. Kemp, A.G. Labouseur, and P.W. Olsen, J. Vijayan, J. Hwang, and J. Yoon. A demonstration of the G^* graph database system. In *IEEE International Conference on Data Engineering – ICDE 2013*, pp. 1356–1359.
- [79] R. E. Tarjan. *Data Structures and Network Algorithms*. SIAM, 1983.
- [80] Transit Feeds. <https://transitfeeds.com>.
- [81] Transport for London. <https://tfl.gov.uk>.
- [82] Δ. Τριάντος. Εφαρμογή Απεικόνισης Βέλτιστων Δρομολογίων MMM σε Έξυπνες Κινητές Συσκευές. Διπλωματική Εργασία, Πανεπιστήμιο Πατρών, 2017.
- [83] E. Tulp and L. Siklossy, TRAINS, an active time-table searcher. *Proc. 8th European Conference on AI*, 1988, pp. 170–175.
- [84] D. Wagner, T. Willhalm, and C. Zaroliagis. Geometric containers for efficient shortest-path computation. *ACM Journal of Experimental Algorithmics*, 10(1.3):1–30, 2005.
- [85] J.W.J. Williams. Algorithm 232 – Heapsort. *Communications of the ACM*, 7(6):347–348, 1964.

- [86] S. Witt. Trip-based public transit routing. In *Algorithms – ESA 2015, Lecture Notes in Computer Science*, Vol. 9294 (Springer 2015), pp. 1025–1036.
- [87] Y. Yang, J.X. Yu, H. Gao, J. Pei, and J. Li. Mining most frequently changing component in evolving graphs. *World Wide Web*, 17:351–376, 2014.
- [88] J. Zhang, X. Long, and T. Suel. Performance of compressed inverted list caching in search engines. In *World Wide Web – WWW 2008*, ACM, pp. 387–396.
- [89] M. Zukowski, S. Héman, N. Nes, and P.A. Boncz. Super-scalar (RAM-CPU) cache compression. In *IEEE International Conference on Data Engineering – ICDE 2006*, pp. 59:1–59:12.