

Αναζήτηση Συντομότερων Μονοπατιών σε
Χρονικά Μεταβαλλόμενα Δίκτυα

Η ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΕΞΕΙΔΙΚΕΥΣΗΣ

υποβάλλεται στην
ορισθείσα από την Γενική Συνέλευση Ειδικής Σύνθεσης
του Τμήματος Πληροφορικής Εξεταστική Επιτροπή

από την

Χρυσούλα Τσιρώνη

ως μέρος των υποχρεώσεων για τη λήψη του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ
ΜΕ ΕΞΕΙΔΙΚΕΥΣΗ
ΣΤΗΝ ΘΕΩΡΙΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Φεβρουάριος 2013



ΒΙΒΛΙΟΘΗΚΗ
ΠΑΝΕΠΙΣΤΗΜΟΥ ΙΩΑΝΝΙΝΩΝ



026000345294



ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΒΙΒΛΙΟΘΗΚΗ

ΔΙΟΡΕΑ:..... Συμφωνία.....
ΑΡΙΘΜΟΣ:..... 10239/2013.....

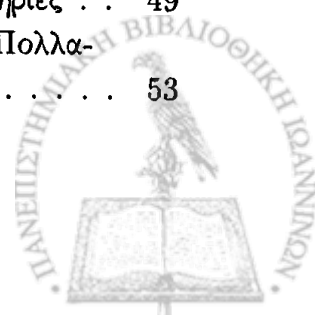
ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να απευθύνω τις ιδιαίτερες ευχαριστίες μου στον επιβλέποντα καθηγητή κ. Σπύρο Κοντογιάννη για την πολύτιμη βοήθειά του στην ολοκλήρωση της παρούσας διατριβής. Η συμβολή του και η καθοδήγησή του ήταν καθοριστική καθώς βασίστηκε στην ουσιώδη και εμβριθή παρουσίαση των τεχνικών λεπτομερειών και εννοιών σχετικά με το αντικείμενο της παρούσας διατριβής.



ΠΕΡΙΕΧΟΜΕΝΑ

1	Εισαγωγή	1
1.1	Εισαγωγή και Στόχοι της Διατριβής	1
1.2	Δομή της Διατριβής	3
2	Χρονικά Μεταβαλλόμενα Συντομότερα Μονοπάτια	5
2.1	Όρισμός του Προβλήματος	5
2.1.1	Κατά τμήματα γραμμικές συναρτήσεις και σημεία καμπής	9
2.1.2	Ο περιορισμός σε FIFO δίκτυα	12
2.2	Επισκόπηση Βιβλιογραφίας για ΧΣΑ	13
2.2.1	Στατικοί αλγόριθμοι	13
2.2.2	Αλγόριθμοι για το πρόβλημα Χρόνου Συντομότερης Άφιξης	15
2.2.3	Αλγόριθμοι για το πρόβλημα Συναρτήσεων Συντομότερης Άφιξης	21
2.3	Παραμετροποιημένα Συντομότερα Μονοπάτια: Ένα Σχετικό Πρόβλημα	21
2.4	Φράγματα Πολυπλοκότητας Συναρτήσεων Συντομότερης Άφιξης	22
2.4.1	Μια πολυεπίπεδη αναπαράσταση του γραφήματος	23
2.4.2	Κάτω φράγμα	24
2.4.3	Άνω φράγμα	26
3	Αλγοριθμικά Αποτελέσματα για Υπολογισμούς Συναρτήσεων Συντομότερης Άφιξης	29
3.1	Αλγόριθμος Χρονικά Εξαρτώμενος από την Έξοδο (output-sensitive)	30
3.1.1	Προεπεξεργασία γραφήματος	30
3.1.2	Περιγραφή του αλγορίθμου	31
3.1.3	Ανάλυση χρονικής πολυπλοκότητας	33
3.2	Προσέγγιση Συναρτήσεων με εκ των Προτέρων Γνώση	33
3.3	Προσεγγιστικός Αλγόριθμος	37
3.4	Προσεγγιστικός Αλγόριθμος για Άγνωστες Συναρτήσεις	42
4	Πολλαπλές αφετηρίες σε Χρονικά Μεταβαλλόμενα Συντομότερα Μονοπάτια	49
4.1	Τεχνικές για Στατικά Συντομότερα Μονοπάτια με Πολλαπλές Αφετηρίες	49
4.2	Επέκταση σε Χρονικά Μεταβαλλόμενα Συντομότερα Μονοπάτια με Πολλαπλές Αφετηρίες	53



5	Συμπεράσματα και Ελεγχτάσεις	57
5.1	Συμπεράσματα και Ελεγχτάσεις	57

1	Εισαγωγή	1
2	Περιγραφή του προβλήματος	2
3	Μεθοδολογία	3
4	Αποτελέσματα	4
5	Συμπεράσματα και Ελεγχτάσεις	57
5.1	Συμπεράσματα και Ελεγχτάσεις	57
5.2	Ελεγχτάσεις	58
5.3	Ελεγχτάσεις	59
5.4	Ελεγχτάσεις	60
5.5	Ελεγχτάσεις	61
5.6	Ελεγχτάσεις	62
5.7	Ελεγχτάσεις	63
5.8	Ελεγχτάσεις	64
5.9	Ελεγχτάσεις	65
5.10	Ελεγχτάσεις	66
5.11	Ελεγχτάσεις	67
5.12	Ελεγχτάσεις	68
5.13	Ελεγχτάσεις	69
5.14	Ελεγχτάσεις	70
5.15	Ελεγχτάσεις	71
5.16	Ελεγχτάσεις	72
5.17	Ελεγχτάσεις	73
5.18	Ελεγχτάσεις	74
5.19	Ελεγχτάσεις	75
5.20	Ελεγχτάσεις	76
5.21	Ελεγχτάσεις	77
5.22	Ελεγχτάσεις	78
5.23	Ελεγχτάσεις	79
5.24	Ελεγχτάσεις	80
5.25	Ελεγχτάσεις	81
5.26	Ελεγχτάσεις	82
5.27	Ελεγχτάσεις	83
5.28	Ελεγχτάσεις	84
5.29	Ελεγχτάσεις	85
5.30	Ελεγχτάσεις	86
5.31	Ελεγχτάσεις	87
5.32	Ελεγχτάσεις	88
5.33	Ελεγχτάσεις	89
5.34	Ελεγχτάσεις	90
5.35	Ελεγχτάσεις	91
5.36	Ελεγχτάσεις	92
5.37	Ελεγχτάσεις	93
5.38	Ελεγχτάσεις	94
5.39	Ελεγχτάσεις	95
5.40	Ελεγχτάσεις	96
5.41	Ελεγχτάσεις	97
5.42	Ελεγχτάσεις	98
5.43	Ελεγχτάσεις	99
5.44	Ελεγχτάσεις	100
5.45	Ελεγχτάσεις	101
5.46	Ελεγχτάσεις	102
5.47	Ελεγχτάσεις	103
5.48	Ελεγχτάσεις	104
5.49	Ελεγχτάσεις	105
5.50	Ελεγχτάσεις	106
5.51	Ελεγχτάσεις	107
5.52	Ελεγχτάσεις	108
5.53	Ελεγχτάσεις	109
5.54	Ελεγχτάσεις	110
5.55	Ελεγχτάσεις	111
5.56	Ελεγχτάσεις	112
5.57	Ελεγχτάσεις	113
5.58	Ελεγχτάσεις	114
5.59	Ελεγχτάσεις	115
5.60	Ελεγχτάσεις	116
5.61	Ελεγχτάσεις	117
5.62	Ελεγχτάσεις	118
5.63	Ελεγχτάσεις	119
5.64	Ελεγχτάσεις	120
5.65	Ελεγχτάσεις	121
5.66	Ελεγχτάσεις	122
5.67	Ελεγχτάσεις	123
5.68	Ελεγχτάσεις	124
5.69	Ελεγχτάσεις	125
5.70	Ελεγχτάσεις	126
5.71	Ελεγχτάσεις	127
5.72	Ελεγχτάσεις	128
5.73	Ελεγχτάσεις	129
5.74	Ελεγχτάσεις	130
5.75	Ελεγχτάσεις	131
5.76	Ελεγχτάσεις	132
5.77	Ελεγχτάσεις	133
5.78	Ελεγχτάσεις	134
5.79	Ελεγχτάσεις	135
5.80	Ελεγχτάσεις	136
5.81	Ελεγχτάσεις	137
5.82	Ελεγχτάσεις	138
5.83	Ελεγχτάσεις	139
5.84	Ελεγχτάσεις	140
5.85	Ελεγχτάσεις	141
5.86	Ελεγχτάσεις	142
5.87	Ελεγχτάσεις	143
5.88	Ελεγχτάσεις	144
5.89	Ελεγχτάσεις	145
5.90	Ελεγχτάσεις	146
5.91	Ελεγχτάσεις	147
5.92	Ελεγχτάσεις	148
5.93	Ελεγχτάσεις	149
5.94	Ελεγχτάσεις	150
5.95	Ελεγχτάσεις	151
5.96	Ελεγχτάσεις	152
5.97	Ελεγχτάσεις	153
5.98	Ελεγχτάσεις	154
5.99	Ελεγχτάσεις	155
5.100	Ελεγχτάσεις	156

ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

2.1	Ένα γράφημα G	10
2.2	Οι συναρτήσεις χρόνου άφιξης για την κάθε ακμή του γραφήματος G του Σχήματος 2.1 και οι αντίστοιχες συνθέσεις των συναρτήσεων αυτών.	11
2.3	Ο χώρος αναζήτησης του μονόδρομου στατικού Dijkstra και ο χώρος αναζήτησης του αμφίδρομου στατικού Dijkstra.	16
2.4	Ο χώρος αναζήτησης του μονόδρομου TD-A* και ο χώρος αναζήτησης του αμφίδρομου TD-ALT.	20
2.5	Ένα γράφημα G και η αντίστοιχη πολυεπίπεδη αναπαράσταση του G . Η πολυεπίπεδη αναπαράσταση δεν έχει ποτέ κύκλους ακόμα και αν το αρχικό γράφημα G έχει κύκλους.	24
2.6	Κάθε μια από τις e_i ακμές ανήκει σε διαφορετικό (s, d) -μονοπάτι.	27
2.7	Το γράφημα $G_{n,c}$ με την πολυεπίπεδη αναπαράστασή του.	27
3.1	Άνω και κάτω φράγματα μιας κατά τμήματα γραμμικής συνάρτησης f που ορίζονται από σχετικό σφάλμα ϵ και από απόλυτο σφάλμα e , αντίστοιχα.	35
3.2	Η βασική ιδέα του Αλγορίθμου 6 με ένα παράδειγμα εκτέλεσης.	36
3.3	Τα άνω και κάτω φράγματα προσέγγισης για την $D(t)$ στο τμήμα $[b', b'']$	39
3.4	Η επαναληπτική διαδικασία υπολογισμού των σημείων της πρώτης φάσης όσο η κλίση της συνάρτησης $D(t)$ είναι τουλάχιστον 1.	41
3.5	Οι τρεις περιπτώσεις για να φράξουμε άνω το απόλυτο σφάλμα μεταξύ δύο διαδοχικών σημείων παρεμβολής. Το άνω φράγμα του απόλυτου σφάλματος παρουσιάζεται από το μπλε ευθύγραμμο τμήμα.	46
4.1	Το όριο της εξωτερικής όψης βρίσκεται στον κύκλο. Ένα δεξιότερο δέντρο συντομότερων μονοπατιών ριζωμένο στην κορυφή r φαίνεται με έντονες ακμές.	51
4.2	Αριστερά φαίνεται ένα γράφημα και ένα σκελετικό δέντρο με έντονες ακμές. Στα δεξιά δείχνουμε το δυϊκό γράφημα που δεν περιέχει τις ακμές που δεν βρίσκονται στο αρχικό δέντρο.	52
4.3	Μια σχηματική αναπαράσταση της συνάρτησης καθυστέρησης $D_{z,s_i}(t)$	55



ΕΥΡΕΤΗΡΙΟ ΑΛΓΟΡΙΘΜΩΝ

1	Στατικός-Dijkstra	14
2	TD-Dijkstra	17
3	TD-Bellman-Ford (label-correcting)	18
4	TD-A*	19
5	TD-ALT	19
6	Τούνελ με φακό	37
7	Σημεία παρεμβολής της $\tilde{f}$ στο $[\tau_0, \tau_N]$	44
8	Διχοτόμηση στο $[c, d]$, BISECT(c, d)	45
9	Στατικά Συντομότερα Μονοπάτια με Πολλαπλές Αφετηρίες	52



ΠΕΡΙΛΗΨΗ

Χρυσούλα Τσιρώνη του Γεωργίου.

Μεταπτυχιακή Εργασία Εξειδίκευσης, Τμήμα Πληροφορικής, Πανεπιστήμιο Ιωαννίνων, Φεβρουάριος 2013.

Αναζήτηση Συντομότερων Μονοπατιών σε Χρονικά Μεταβαλλόμενα Δίκτυα

Επιβλέπων: Σπύρος Κοντογιάννης.

Στη παρούσα διατριβή μελετάται το πρόβλημα εύρεσης συντομότερων μονοπατιών σε χρονικά μεταβαλλόμενα δίκτυα. Αναλύουμε τις θεωρητικές και δομικές ιδιότητες του προβλήματος και παρουσιάζουμε ορισμένες αλγοριθμικές τεχνικές που αντιμετωπίζουν το πρόβλημα. Τα χρονικά μεταβαλλόμενα δίκτυα, στα οποία η συνάρτηση χρόνου κατά μήκος μιας ακμής είναι μια γνωστή συνάρτηση του χρόνου αναχώρησης από το ένα άκρο της ακμής, εμφανίζονται σε αρκετές πρακτικές εφαρμογές που είναι σχετικές με την δρομολόγηση αιτημάτων σε κοινωνικά δίκτυα, δίκτυα τηλεπικοινωνιών, και άλλες παρεμφερή περιοχές. Καθώς το γενικό πρόβλημα είναι NP-πλήρες, επικεντρώνουμε το ενδιαφέρον μας στην περίπτωση των FIFO δικτύων, στα οποία τα προϊόντα ταξιδεύουν κατά μήκος μιας ακμής με έναν τρόπο που σέβεται την First-In-First-Out ιδιότητα. Η ειδική αυτή περίπτωση εμφανίζεται συχνά στην πράξη και μας επιτρέπει την τεκμηρίωση θεωρητικών ιδιοτήτων και την σχεδίαση αποτελεσματικών αλγορίθμων. Μελετάμε σχετικά προβλήματα όπως τα παραμετροποιημένα συντομότερα μονοπάτια και την προσέγγιση άγνωστων συναρτήσεων. Παρουσιάζουμε άνω και κάτω γνωστά φράγματα για το βασικό πρόβλημα όπως επίσης και γνωστούς πολυωνυμικού χρόνου προσεγγιστικούς αλγορίθμους. Επίσης, βασισμένοι σε προηγούμενα και καινούρια αποτελέσματα, σχεδιάζουμε έναν αποτελεσματικό αλγόριθμο για να εξασφαλίσει κανείς συντομότερα χρονικά μεταβαλλόμενα μονοπάτια με πολλαπλές αφετηρίες. Σκοπός της παρούσας διατριβής είναι να παρουσιάσουμε ένα γενικό μοτίβο τεχνικών που έχουν αναπτυχθεί για διάφορες παραλλαγές του προβλήματος.



EXTENDED ABSTRACT IN ENGLISH

Chrysoula Tsironi G.

MSc, Computer Science Department, University of Ioannina, Greece. February 2013.

Shortest Paths in Time-Dependent Networks

Thesis Supervisor: Spyros Kontogiannis.

In this thesis we present a concise study of the time-dependent shortest path problem, its theoretical properties, and its algorithmic solutions. Time-dependent networks, in which the travel time along each arc is a known function of the departure time along the arc, arise in many practical applications, particularly those related to vehicular transportation, social networks, telecommunication networks and other related areas. Since the general problem is NP-hard, we focus entirely on the case of FIFO networks, in which commodities travel through arcs in a First-In-First-Out manner. This special case is very common in practice and enables the development of rich theoretical properties and efficient solution algorithms. We study relevant problems such as the parametric shortest paths and approximation of unknown functions. Further we give already known upper and lower bounds for the general time-dependent problem as well as known polynomial-time approximation algorithms. Moreover based on previous and new results we design an efficient algorithm for the time-dependent shortest paths with multiple sources. Our aim in this thesis is to present a unified framework which encompasses a wide range of problem variants and ties together past work and recent developments.



ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Εισαγωγή και Στόχοι της Διατριβής

1.2 Δομή της Διατριβής

1.1 Εισαγωγή και Στόχοι της Διατριβής

Ένα από τα βασικά προβλήματα στη σχεδίαση δικτύων είναι η εύρεση του συντομότερου μονοπατιού μεταξύ μιας πηγής και ενός προορισμού. Το πρόβλημα αυτό βρίσκει ποικίλες εφαρμογές σε θέματα δρομολόγησης και βελτιστοποίησης ροής όπου εμπλέκονται προβλήματα μετακίνησης. Η σχεδίαση και ανάλυση αποτελεσματικών αλγοριθμικών τεχνικών για τέτοιου είδους προβλήματα αποσκοπεί στον γρήγορο υπολογισμό του συντομότερου μονοπατιού. Παρά την εκτεταμένη μελέτη σε προβλήματα όπου η καθυστέρηση κατά την κίνηση μεταξύ σημείων θεωρείται ως μια στατική συνάρτηση, τα τελευταία χρόνια ιδιαίτερη έμφαση δίνεται σε δίκτυα όπου η καθυστέρηση μετακίνησης εκφράζεται ως μια συνάρτηση του χρόνου.

Ωστόσο, στην πράξη ένα δίκτυο μπορεί να εκφράζεται με δυναμικά χαρακτηριστικά που απαιτούν ιδιαίτερες προσεγγίσεις για την εύρεση συντομότερου μονοπατιού. Δύο συχνές προσεγγίσεις υπάρχουν για προβλήματα συντομότερων μονοπατιών με δυναμικά χαρακτηριστικά. Στη πρώτη προσέγγιση κάποιος πρέπει να επανυπολογίσει συντομότερα μονοπάτια λόγω συχνών απρόβλεπτων αλλαγών στα δεδομένα του δικτύου. Αυτή η κατηγορία αποτελεί στην ουσία ένα πρόβλημα επαναβελτιστοποίησης για την επίλυση μιας διαδοχικής ακολουθίας από παραλλαγές προβλημάτων του στατικού συντομότερου μονοπατιού. Η δεύτερη προσέγγιση, που θα μελετήσουμε εδώ, είναι το πρόβλημα του χρονικά μεταβαλλόμενου συντομότερου μονοπατιού όπου τα χαρακτηριστικά του δικτύου μεταβάλλονται ως προς τον χρόνο με προκαθορισμένο τρόπο. Τέτοιου είδους προβλήματα εμφανίζονται συχνά στην πράξη, όπως για παράδειγμα στη μετακίνηση οχημάτων. Το συντομότερο μονοπάτι που υπολογίστηκε κάποια χρονική στιγμή με τα συγκεκριμένα δεδομένα του δικτύου μπορεί

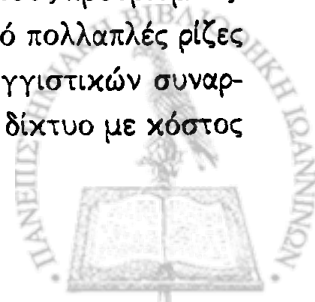


να μην παραμείνει βέλτιστο όταν εξετάζουμε προκαθορισμένες αλλαγές ως προς το χρόνο στις ακμές του δικτύου που επιβάλλονται να εμφανιστούν καθώς κάποιος μετακινείται στο δίκτυο. Κλασικό παράδειγμα στα μέσα μεταφοράς ενός δικτύου είναι οι χρονικές περιόδους κοντά στις ώρες αιχμής όπου συμβαίνουν πολλές τέτοιες αλλαγές συντομότερων μονοπατιών.

Στο πρόβλημα του χρονικά μεταβαλλόμενου συντομότερου μονοπατιού θεωρούμε ότι ο χρόνος μεταφοράς κατά μήκος μιας ακμής περιγράφεται από μια συνάρτηση του χρόνου αναχώρησης από το ένα άκρο της ακμής, και όλες αυτές οι συναρτήσεις χρόνου είναι γνωστές εκ των προτέρων για κάθε χρονική στιγμή. Το πρόβλημα αυτό είχε αρχικά εισαχθεί από τους Cook και Halsey [5] το 1966 όταν είχε προταθεί με τον περιορισμό ότι ο χρόνος παίρνει διακριτές τιμές.

Το γενικό πρόβλημα του χρονικά μεταβαλλόμενου συντομότερου μονοπατιού είναι NP-δύσκολο ως προς την πολυπλοκότητά του, καθώς σε αυτό ανάγονται μια σειρά από NP-δύσκολα προβλήματα, όπως είναι το πρόβλημα του σακιδίου. Ωστόσο ανάλογα με τον τρόπο που ορίζει κανείς το πρόβλημα μπορεί να μην ανήκει ούτε στην κλάση NP, καθώς η έξοδος του προβλήματος μπορεί να μην φράσσεται από μια πολυωνυμική συνάρτηση. Επιπλέον, όπως έδειξαν οι Orda και Rom [32], υπάρχουν στιγμιότυπα του προβλήματος με πεπερασμένο ελάχιστο χρόνο μετάβασης, όπου όμως όλα τα βέλτιστα (προφανώς όχι απλά) μονοπάτια απαρτίζονται από άπειρες εμφανίσεις ακμών. Εμείς θα θεωρήσουμε ότι οι τα δίκτυα που μελετάμε σέβονται τη FIFO ιδιότητα, δηλαδή τα δεδομένα του δικτύου ταξιδεύουν κατά μήκος μιας ακμής με τρόπο First-in-First-out. Στην πραγματικότητα πολλά δίκτυα εφαρμογών παρουσιάζουν τη FIFO ιδιότητα όπως είναι ορισμένα κλασικά δίκτυα μεταφορών (τρένα μεταφοράς), τα οδικά δίκτυα όπου υπάρχουν όρια ταχύτητας, τα κοινωνικά δίκτυα όπου οι σύνδεσμοι εκφράζουν βαθμό επιρροής, κ.τ.λ..

Με τη FIFO ιδιότητα το πρόβλημα του χρονικά μεταβαλλόμενου μονοπατιού παρουσιάζει πολύ ενδιαφέρουσες ιδιότητες που μας επιτρέπουν την σχεδίαση αποτελεσματικών πολυωνυμικού-χρόνου αλγορίθμους για την επίλυσή του. Υπό αυτό το πρίσμα, μελετάμε σχετικά προβλήματα όπως τα παραμετροποιημένα συντομότερα μονοπάτια και την προσέγγιση ελάχιστων τιμών που περιγράφονται από άγνωστες κατά-τμήματα γραμμικές συναρτήσεις. Μια τέτοιου είδους συνάρτηση που εκφράζει την χρονική μεταβολή παρουσιάζει ιδιαίτερο ενδιαφέρον καθώς μπορούμε να εφαρμόσουμε μια περιοδική δειγματοληψία των χρόνων μετάβασης σε ακμές για συγκεκριμένες στιγμές αναχώρησης, και στη συνέχεια να θεωρήσουμε ως (συνεχή) συνάρτηση μετάβασης την κατά τμήματα γραμμική συνάρτηση που προκύπτει από την γραμμική παρεμβολή (interpolation) των σημείων δειγματοληψίας. Θα παρουσιάσουμε άνω και κάτω γνωστά υπολογιστικά φράγματα για το πρόβλημα, όπως επίσης γνωστούς προσεγγιστικούς αλγορίθμους πολυωνυμικού χρόνου. Επίσης παρέχουμε αναλυτική παρουσίαση ενός αλγορίθμου που υπολογίζει σε πολυωνυμικό χρόνο προσεγγιστικά συναρτήσεις καθυστέρησης από μια αφετηρία προς όλους τους δυνατούς προορισμούς. Υιοθετώντας μια γνωστή ιδέα από την εύρεση ελαχίστων αποστάσεων από πολλαπλές ρίζες σε στατικά δίκτυα [26], παρουσιάζουμε μια τεχνική υπολογισμού προσεγγιστικών συναρτήσεων απόστασης από πολλαπλές αφετηρίες σε χρονικά μεταβαλλόμενο δίκτυο με κόστος



ασυμπτωτικά ίσο με αυτό του υπολογισμού από μια αφετηρία προς όλους τους προορισμούς. Σκοπός της παρούσας διατριβής είναι να παρουσιάσουμε ένα γενικό μοτίβο τεχνικών που έχουν αναπτυχθεί για διάφορες παραλλαγές του προβλήματος.

1.2 Δομή της Διατριβής

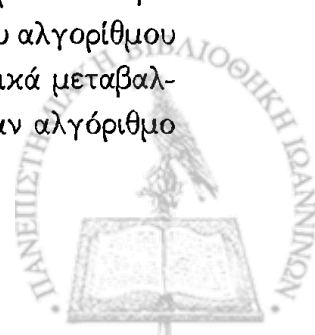
Η διατριβή περιέχει 5 κεφάλαια: Το Κεφάλαιο 1 εισάγει το πρόβλημα του χρονικά μεταβαλλόμενου συντομότερου μονοπατιού, αναδεικνύει τον σκοπό της διατριβής και παρουσιάζει την δομή που ακολουθήσαμε στην εργασία.

Στο Κεφάλαιο 2 ορίζουμε το κλασικό στατικό πρόβλημα και δυο βασικά προβλήματα σε χρονικά μεταβαλλόμενο περιβάλλον που θα μελετήσουμε:

- α) Υπολογισμός του χρόνου (=τιμής) συντομότερης άφιξης στον (στους) προορισμό (-ούς) δεδομένου του χρόνου αναχώρησης από την αφετηρία.
- β) Περιγραφή της (των) συνάρτησης (-ων) συντομότερων αφίξεων στον (στους) προορισμό (-ούς) για κάθε χρονική στιγμή από την (κοινή) αφετηρία.

Αξίζει να σημειώσουμε ότι η δεύτερη εκδοχή του προβλήματος μπορεί να οδηγήσει και στην προσέγγιση άγνωστων συναρτήσεων. Δίνουμε μια σύντομη περιγραφή των δεδομένων εισόδου βάσει των οποίων προκύπτουν οι κατά τμήματα γραμμικές συναρτήσεις για την περιγραφή των καθυστερήσεων στις ακμές. Επίσης εισάγουμε βασικές έννοιες όπως τα σημεία καμπής που εμφανίζονται κατά την σύνθεση συναρτήσεων χρόνου άφιξης και δικαιολογούμε τον περιορισμό σε FIFO δίκτυα. Παραθέτουμε κλασικούς αλγορίθμους από την βιβλιογραφία για το στατικό πρόβλημα, όπως είναι ο αλγόριθμος του Dijkstra και αναλύουμε την αντίστοιχη εκδοχή τους σε χρονικά μεταβαλλόμενο περιβάλλον τόσο για διακριτές τιμές όσο και για συνεχείς τιμές του χρόνου. Ειδικότερα για διακριτές τιμές θα παρουσιάσουμε τον αλγόριθμο TD-Dijkstra που επιλύει την περίπτωση α) και τον αλγόριθμο TD-Bellman-Ford που επιλύει την περίπτωση β), αντίστοιχα. Επιπρόσθετα για την περίπτωση α) θα αναφερθούμε σε επιταχύνσεις του κλασικού αλγορίθμου του Dijkstra όπως είναι ο αλγόριθμος A^* και ο αλγόριθμος ALT και θα περιγράψουμε τις αντίστοιχες παραλλαγές τους TD- A^* και TD-ALT σε χρονικά μεταβαλλόμενα δίκτυα. Σημειώνουμε ότι το υπόλοιπο κομμάτι της διατριβής θα επικεντρωθεί στην περίπτωση β) για συνεχείς τιμές του χρόνου. Δείχνουμε την σύνδεση του προβλήματος με το πρόβλημα των παραμετροποιημένων συντομότερων μονοπατιών και κλείνουμε το κεφάλαιο αυτό με την διατύπωση άνω και κάτω υπολογιστικών φραγμάτων ως προς το πλήθος των σημείων καμπής.

Το Κεφάλαιο 3 περιλαμβάνει αλγοριθμικά αποτελέσματα για πλήρη περιγραφή συναρτήσεων συντομότερη άφιξης όταν οι καθυστερήσεις ακμών περιγράφονται από κατά-τμήματα γραμμικές συναρτήσεις που υπακούνε τη FIFO ιδιότητα. Πρώτα παρουσιάζουμε ένα αλγόριθμο χρονικά εξαρτώμενο από την έξοδο δηλαδή η χρονική πολυπλοκότητα του αλγορίθμου εξαρτάται από το πλήθος των σημείων καμπής που εμφανίζονται σε ένα χρονικά μεταβαλλόμενο δίκτυο. Πριν από τον προσεγγιστικό αλγόριθμο θα περιγράψουμε έναν αλγόριθμο



που προσεγγίζει με ελάχιστο πλήθος σημείων καμπής μια κατά τμήματα γραμμική συνάρτηση έχοντας, ωστόσο, γνώση της συνάρτησης χρόνου μετάβασης. Στη συνέχεια θα δείξουμε έναν πολυωνυμικού-χρόνου εκτέλεσης προσεγγιστικό αλγόριθμο που κατασκευάζει μια προσέγγιση (μικρού μεγέθους) για την συνάρτηση του χρόνου μετάβασης.

Στο Κεφάλαιο 4 μελετάμε την εύρεση συντομότερων μονοπατιών από ένα σύνολο πολλαπλών αφετηριών προς όλες τις προσβάσιμες από αυτές κορυφές. Στόχος είναι ο συγκεκριμένος υπολογισμός να κοστίζει ασυμπτωτικά όσο και ο υπολογισμός από μια μόνο αφετηρία. Αναφερόμαστε πρώτα σε έναν γνωστό αλγόριθμο για το στατικό πρόβλημα που επιλύει με αποτελεσματικό και γρήγορο τρόπο σε επίπεδα γραφήματα όπου οι πολλαπλές αφετηρίες βρίσκονται στην εξωτερική όψη του γραφήματος. Στη συνέχεια δείχνουμε έναν αποτελεσματικό αλγόριθμο για να επεκτείνουμε την ιδέα αυτή σε χρονικά μεταβαλλόμενο περιβάλλον χωρίς ωστόσο να περιοριζόμαστε σε επίπεδα γραφήματα και στην συγκεκριμένη τοποθέτηση για τις αφετηρίες.

Ολοκληρώνουμε την διατριβή με το Κεφάλαιο 5 συνοψίζοντας τις τεχνικές που παρουσιάσαμε για χρονικά μεταβαλλόμενα συντομότερα μονοπάτια και παραθέτουμε ορισμένα ανοιχτά προβλήματα και επεκτάσεις για πιθανή μελλοντική μελέτη.



ΚΕΦΑΛΑΙΟ 2

ΧΡΟΝΙΚΑ ΜΕΤΑΒΑΛΛΟΜΕΝΑ ΣΥΝΤΟΜΟΤΕΡΑ ΜΟΝΟΠΑΤΙΑ

2.1 Ορισμός του Προβλήματος

2.2 Επισκόπηση Βιβλιογραφίας για ΧΣΑ

2.3 Παραμετροποιημένα Συντομότερα Μονοπάτια: Ένα Σχετικό Πρόβλημα

2.4 Φράγματα Πολυπλοκότητας Συναρτήσεων Συντομότερης Άφιξης

2.1 Ορισμός του Προβλήματος

Συμβολίζουμε με $G = (V, E)$ ένα κατευθυνόμενο γράφημα με $V = V(G)$ το σύνολο κορυφών και $E = E(G) \subseteq V(G) \times V(G)$ το σύνολο ακμών του γραφήματος G . Για το πλήθος των κορυφών θα χρησιμοποιούμε $n = |V(G)|$ και για το πλήθος των ακμών θα χρησιμοποιούμε $m = |E(G)|$. Μια ακμή συμβολίζεται ως $e = (u, v)$, ή $\overrightarrow{(u, v)}$ δηλώνοντας ως u την πηγή (ή την ουρά) της ακμής και v την άφιξη (ή την κεφαλή) της ακμής. Τότε θα λέμε ότι η κορυφή u είναι προσκείμενη στην εξερχόμενη ακμή (u, v) ενώ ανάλογα η ακμή (u, v) είναι εισερχόμενη ακμή για την κορυφή v . Δοθέντος ενός γραφήματος $G = (V, E)$ αν αντιστρέψουμε τις κατευθύνσεις των ακμών τότε έχουμε το ανεστραμμένο γράφημα που το συμβολίζουμε με $\overleftarrow{G}$.

Το γράφημα $G = (V, E)$ με $V(G) = \{v_1, \dots, v_n\}$ και $\{v_i, v_{i+1}\} \in E(G)$ καλείται μονοπάτι. Τα άκρα του μονοπατιού είναι οι κορυφές v_1, v_n και οι υπόλοιπες κορυφές καλούνται εσωτερικές. Ένα (x, y) -μονοπάτι ενός γραφήματος είναι μονοπάτι με άκρα τις κορυφές x και y του γραφήματος. Για τα άκρα x, y δηλώνουμε ως $\mathcal{P}_{x,y}$ το σύνολο όλων των (x, y) -μονοπατιών.



Θεωρούμε επίσης ότι υπάρχει μια συνάρτηση $w : E \rightarrow \mathbb{R}$. Με άλλα λόγια η συνάρτηση w περιγράφει (στατικές) καθυστερήσεις στις ακμές. Το *συντομότερο μονοπάτι* του γραφήματος G μεταξύ δύο κορυφών s και d είναι εκείνο το (s, d) -μονοπάτι με το ελάχιστο άθροισμα των βαρών. Η εύρεση του συντομότερου μονοπατιού μεταξύ των κορυφών s (*αφετηρία*) και d (*προορισμός*) αποτελεί το πρόβλημα που περιγράφεται ακολούθως.

ΣΥΝΤΟΜΟΤΕΡΟ ΜΟΝΟΠΑΤΙ ΑΦΕΤΗΡΙΑΣ-ΠΡΟΟΡΙΣΜΟΥ (ΣΜΑΠ)

Είσοδος: Ένα γράφημα $G = (V, E)$, $w : E \rightarrow \mathbb{R}$ και δύο κορυφές $s, d \in V(G)$

Ερώτημα: Εύρεση του (s, d) -μονοπατιού $P \in \mathcal{P}_{s,d}$ τέτοιο ώστε

$$\sum_{e \in P} w(e) = \min_{Q \in \mathcal{P}_{s,d}} \sum_{e \in Q} w(e).$$

Η παραπάνω περιγραφή αποτελεί την στατική εκδοχή του προβλήματος. Αν δεν έχουμε βάρη στις ακμές τότε μπορεί κανείς να θεωρήσει ότι $w : E \rightarrow \{c\}$ και $c \in \mathbb{R}^+$ κάποια θετική σταθερά, όπου τότε ψάχνουμε για το (s, d) -μονοπάτι με το ελάχιστο πλήθος ακμών. Όπως θα δούμε αργότερα υπάρχουν διάφορες παραλλαγές και γενικεύσεις του προβλήματος. Το προηγούμενο πρόβλημα αναφέρεται συχνά ως *συντομότερο μονοπάτι μεταξύ αφετηρίας-προορισμού* (ΣΜΑΠ) για να ξεχωρίσει από τις ακόλουθες εκδοχές:

- Το πρόβλημα **ΣΥΝΤΟΜΟΤΕΡΑ ΜΟΝΟΠΑΤΙΑ ΑΠΟ ΑΦΕΤΗΡΙΑ (ΣΜΑ)**, στο οποίο θέλουμε να βρούμε τα συντομότερα μονοπάτια από μια διακεκριμένη κορυφή (*αφετηρία*) s προς όλες τις υπόλοιπες κορυφές του γραφήματος.
- Το πρόβλημα **ΣΥΝΤΟΜΟΤΕΡΑ ΜΟΝΟΠΑΤΙΑ ΠΡΟΣ ΠΡΟΟΡΙΣΜΟ (ΣΜΠ)**, στο οποίο θέλουμε να βρούμε τα συντομότερα μονοπάτια από όλες τις κορυφές ενός γραφήματος προς μια διακεκριμένη κορυφή (*προορισμός*) d . Το πρόβλημα αυτό ανάγεται στο πρόβλημα **ΣΥΝΤΟΜΟΤΕΡΑ ΜΟΝΟΠΑΤΙΑ ΑΠΟ ΑΦΕΤΗΡΙΑ** αν αντιστρέψουμε τις κατευθύνσεις των ακμών του γραφήματος.
- Το πρόβλημα **ΣΥΝΤΟΜΟΤΕΡΑ ΜΟΝΟΠΑΤΙΑ ΓΙΑ ΚΑΘΕ ΖΕΥΓΟΣ ΑΦΕΤΗΡΙΑΣ – ΠΡΟΟΡΙΣΜΟΥ (ΣΜΖ)**, στο οποίο σκοπός μας είναι η εύρεση των συντομότερων μονοπατιών μεταξύ κάθε ζεύγους κορυφών u, v του γραφήματος.

Το πρόβλημα του συντομότερου μονοπατιού (που θα το αναφέρουμε και ως **ΣΤΑΤΙΚΟ ΣΥΝΤΟΜΟΤΕΡΟ ΜΟΝΟΠΑΤΙ**) έχει μελετηθεί διεξοδικά από την σκοπιά της αλγοριθμικής θεωρίας γραφημάτων (π.χ., [1, 9, 13]). Για το στατικό πρόβλημα, υπάρχουν κι άλλες τεχνικές (κυρίως ευρετικές) που αν και θεωρητικά δε βελτιώνουν το Dijkstra, στην πράξη είναι πολύ ταχύτερες, όπως για παράδειγμα οι αλγόριθμοι A^* , Arc Flags, Contraction Hierarchies, και ALT [16, 18, 20, 33]. Η βασική ιδέα είναι να γίνει προεπεξεργασία των δεδομένων και να παραχθεί (στατικά) κάποια βοηθητική πληροφορία (ανεξάρτητα από την αφετηρία και τον προορισμό που μας ενδιαφέρει) που επιτρέπουν τη σημαντική επιτάχυνση του Dijkstra, για οποιοδήποτε ζεύγος αφετηρίας – προορισμού κι αν ζητήσουμε σε πραγματικό χρόνο. Για τον λόγο αυτό ονομάζονται *speedup techniques*. Υπάρχουν επίσης οι *χρησμοί αποστάσεων* (Distance Oracles), που κάνουν κάτι ανάλογο με τις προηγούμενες

τεχνικές, όμως αποδεδειγμένα καταφέρνουν πολύ καλό χρόνο απόκρισης (query time) σε οποιοδήποτε ερώτημα ΣΜΑΠ, με αποδεκτό (πολυωνυμικό) χρόνο προεπεξεργασίας για τη δημιουργία των βοηθητικών δεδομένων [12]. Και οι δυο παραπάνω κατηγορίες δουλεύουν ιδιαίτερα καλά σε αραιά (πχ, σε επίπεδα, αλλά όχι μόνο) γραφήματα.

Ωστόσο στην πράξη πολλά δίκτυα τείνουν να έχουν δυναμικά χαρακτηριστικά που απαιτούν διαφορετικές στρατηγικές για τον υπολογισμό συντομότερων μονοπατιών. Υπάρχουν δύο τύποι δυναμικών συντομότερων μονοπατιών: στον πρώτο τύπο ανήκουν τα προβλήματα στα οποία πρέπει να επανυπολογιστούν τα συντομότερα μονοπάτια λόγω συχνών, στιγμιαίων και απρόβλεπτων αλλαγών στα δεδομένα του δικτύου. Αυτός ο τύπος προβλημάτων αφορά ουσιαστικά την επίλυση μιας ακολουθίας από στενά συσχετιζόμενα στατικά προβλήματα. Ο δεύτερος τύπος προβλημάτων, που αποτελεί τον κορμό της εργασίας αυτής, είναι το πρόβλημα των *χρονικά μεταβαλλόμενων συντομότερων μονοπατιών* στα οποία τα χαρακτηριστικά του δικτύου αλλάζουν συναρτήσει του χρόνου με προκαθορισμένο τρόπο.

Τέτοια προβλήματα εμφανίζονται συχνά σε συγκοινωνιακά ή τηλεπικοινωνιακά δίκτυα όπου ο κυκλοφοριακός φόρτος εξαρτάται από τη χρονική στιγμή της χρήσης κάθε συνδέσμου: το συντομότερο μονοπάτι που έχει υπολογιστεί σε ένα στιγμιότυπο του δικτύου κάποια χρονική στιγμή μπορεί να μην είναι βέλτιστο αν θεωρήσουμε προκαθορισμένες μελλοντικές αλλαγές στην μετακίνηση καθυστέρησης κατά μήκος μιας ακμής, που θα εμφανιστούν ιδιαίτερα σε χρονικές περιόδους όπως είναι οι ώρες αιχμής. Στο πρόβλημα των *χρονικά μεταβαλλόμενων συντομότερων μονοπατιών* θεωρούμε ότι ο χρόνος μετακίνησης κατά μήκος μιας ακμής είναι μια συνάρτηση του χρόνου αναχώρησης από την ουρά της ακμής και οι συναρτήσεις αυτές είναι δοσμένες (γνωστές) εξ αρχής για όλη την διάρκεια του χρόνου. Το πρόβλημα αυτό εμφανίστηκε το 1966 όταν προτάθηκε πρώτη φορά για διακριτό χρόνο από τους Cooke και Halsey [5]. Στη συνέχεια περιγράφουμε αναλυτικά το πρόβλημα των *χρονικά μεταβαλλόμενων συντομότερων μονοπατιών*.

Θεωρούμε ότι ο χρόνος έχει θετικές τιμές, δηλαδή για κάθε χρονική στιγμή $t \in \mathbb{R}^+$ και οποιαδήποτε συνάρτηση υπολογισμού κάποιας χρονικής στιγμής είναι μια συνεχής συνάρτηση. Μια συνάρτηση χρόνου άφιξης ακμής $e \in E(G)$ δέχεται ως είσοδο μια χρονική στιγμή $t \in \mathbb{R}^+$ και συμβολίζεται ως $A_e(t)$. Η τιμή $A_e(t)$ δηλώνει την χρονική στιγμή άφιξης στην κορυφή $\text{head}(e)$ αν κάποιος αναχωρήσει από την κορυφή $\text{tail}(e)$ την χρονική στιγμή t . Σε ένα γράφημα $G = (V, E)$ όπου για κάθε ακμή υπάρχει μια συνάρτηση χρόνου άφιξης A_e τότε η συνάρτηση για την ακμή e περιγράφεται ως $A_e : \mathbb{R}^+ \rightarrow \mathbb{R}^+$. Ο *χρόνος μετάβασης* δια μέσου της ακμής (i, j) ξεκινώντας από την κορυφή i την χρονική στιγμή t περιγράφεται από την συνάρτηση $D_{(i,j)}(t) = A_{(i,j)}(t) - t$.

Ορισμός 2.1. Αν η συνάρτηση χρόνου άφιξης $A_e(t)$ είναι μη-φθίνουσα τότε θα λέμε ότι η ακμή e ακολουθεί την *FIFO ιδιότητα*. Δηλαδή Επομένως για δύο χρονικές στιγμές $t \leq t'$ θα ισχύει $A_e(t) + t \leq A_e(t') + t'$.

Δοθέντος ενός (u_1, u_k) -μονοπατιού P που περιγράφεται ως $P = (u_1, u_2, \dots, u_k)$, η συνάρτηση χρόνου άφιξης μονοπατιού συμβολίζεται ως $A[P](t)$ και αποτελείται από την σύνθεση επιμέρους συναρτήσεων χρόνου άφιξης ως εξής:

$$A[P](t) = A_{(u_{k-1}, u_k)} (A_{(u_{k-2}, u_{k-1})} (\dots A_{(u_1, u_2)}(t))) .$$



Επειδή αποτελείται από την σύνθεση μη-φθίνουσων συναρτήσεων, έπεται με απλή επαγωγή ως προς το πλήθος των ακμών ότι οι συναρτήσεις χρόνου άφιξης μονοπατιών είναι μη-φθίνουσες. Επομένως σε ένα γράφημα (δίκτυο) G με συναρτήσεις χρόνου άφιξης που υπαχούν την FIFO ιδιότητα θα λέμε ότι το G είναι ένα FIFO δίκτυο. Σημειώνουμε ότι η FIFO ιδιότητα αφορά τις συναρτήσεις χρόνου άφιξης ακμών. Όμως, με ένα απλό επίσης επαγωγικό επιχείρημα μπορεί κανείς να δείξει ότι αν όλες οι ακμές υπαχούν την FIFO ιδιότητα τότε κάθε μονοπάτι του γραφήματος υπακούει την FIFO ιδιότητα. Κάθε οντότητα σε ένα FIFO δίκτυο ταξιδεύει δια μέσω ακμών με τρόπο που σέβεται την First-in-First-out ιδιότητα. Δηλαδή η ισχύς της FIFO ιδιότητας για δίκτυα σημαίνει πρακτικά ότι δεν υπάρχει λόγος αναμονής σε ενδιάμεσους κόμβους, καθώς κινούμαστε από την αφετηρία προς τον προορισμό. Στο εξής θεωρούμε, εκτός αν δηλώνεται ρητά διαφορετικά, ότι για κάθε γράφημα G που μελετάμε σέβεται την FIFO ιδιότητα.

Ορισμός 2.2. Η συνάρτηση συντομότερης άφιξης συμβολίζεται με $A[s, d](t)$ για την αφετηρία s και τον προορισμό d και ορίζεται ως η ελάχιστη τιμή $A[P](t)$ για κάθε (s, d) -μονοπάτι P του συνόλου $\mathcal{P}_{s,d}$. Με άλλα λόγια η συνάρτηση $A[s, d](t)$ δίνει την συντομότερη άφιξη στον προορισμό d αν κανείς αναχωρήσει από την αφετηρία s την χρονική στιγμή t .

Επειδή η συνάρτηση $A[s, d](t)$ είναι η ελάχιστη τιμή από μη-φθίνουσες συναρτήσεις, έπεται ότι και η $A[s, d](t)$ είναι μη-φθίνουσα. Επίσης από τον ορισμό της FIFO ιδιότητας, οι μερικές παράγωγοι κάθε συνάρτησης A_e είναι μη-αρνητικές συναρτήσεις. Δηλαδή $A[s, d](t) \geq t$ για κάθε χρονική στιγμή t . Θα λέμε ότι η συνάρτηση ελάχιστης καθυστέρησης από το s στο d δίνεται από τον τύπο $A[s, d](t) - t$.

Το πρόβλημα του χρονικά μεταβαλλόμενου συντομότερου μονοπατιού έγκειται στον υπολογισμό της συνάρτησης συντομότερης άφιξης στον προορισμό d αν κανείς αναχωρήσει από τον s κάποια δεδομένη χρονική στιγμή t_0 . Πιο συγκεκριμένα το πρόβλημα αυτό ορίζεται ως εξής:

ΧΡΟΝΟΙ ΣΥΝΤΟΜΟΤΕΡΗΣ ΑΦΙΞΗΣ, ΓΙΑ ΔΕΔΟΜΕΝΗ ΑΝΑΧΩΡΗΣΗ ΑΠΟ ΑΦΕΤΗΡΙΑ (ΧΣΑ)

Είσοδος: Ένα γράφημα $G = (V, E)$, μια συνάρτηση $A : E \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$, $s, d \in V(G)$, και $t_0 \in \mathbb{R}^+$

Ερώτημα: Υπολογισμός της τιμής $A[s, d](t_0)$ και εύρεση του (s, d) -μονοπατιού $P \in \mathcal{P}_{s,d}$ τέτοιο ώστε $A[P](t_0) = \min_{Q \in \mathcal{P}_{s,d}} A[Q](t_0)$.

Μπορεί κανείς να αναρωτηθεί για την περιγραφή ολόκληρης της συνάρτησης $A[s, d]$ και τον υπολογισμό συντομότερων μονοπατιών σε χρονικά μεταβαλλόμενα δίκτυα χωρίς κάποιο προκαθορισμένο χρόνο αναχώρησης από την αφετηρία s . Στην ουσία, δηλαδή, σκοπός είναι η πλήρης περιγραφή της συνάρτησης συντομότερης άφιξης $A[s, d](t)$, για οποιαδήποτε χρονική στιγμή αναχώρησης από την αφετηρία. Με την επίτευξη του στόχου μπορεί κανείς να σχεδιάσει έναν χάρτη ταξιδιού που υπολογίζει για οποιαδήποτε επιθυμητό χρόνο άφιξης, τον βέλτιστο χρόνο αναχώρησης με σκοπό την ελαχιστοποίηση του συνολικού χρόνου ταξιδιού. Συμπληρωματικά επιθυμούμε να βρούμε και το εκάστοτε μονοπάτι (για οποιαδήποτε

χρονική στιγμή αναχώρησης από την αφετηρία) σε χρόνο που να εξαρτάται (π.χ., γραμμικά) από το ίδιο το μονοπάτι που πρέπει να επιστραφεί και όχι από το μέγεθος ή την τοπολογία του δικτύου. Τότε το συγκεκριμένο πρόβλημα μπορεί να διατυπωθεί ως εξής:

ΣΥΝΑΡΤΗΣΕΙΣ ΣΥΝΤΟΜΟΤΕΡΗΣ ΑΦΙΞΗΣ, ΓΙΑ ΑΝΑΧΩΡΗΣΗ ΑΠΟ ΔΕΔΟΜΕΝΗ ΑΦΕΤΗΡΙΑ (ΣΣΑ)

Είσοδος: Ένα γράφημα $G = (V, E)$, μια συνάρτηση $A : E \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$, και $s, d \in V(G)$

Ερώτημα: (i) Περιγραφή (π.χ., μέσω των σημείων καμπής) της συνάρτησης $A[s, d]$ και
(ii) Προσδιορισμός μονοπατιού ελάχιστου μήκους για κάθε χρονική στιγμή αναχώρησης, σε χρόνο γραμμικό ως προς το μήκος του μονοπατιού.

Όταν θέλουμε να αναφερθούμε και στα δύο προβλήματα τότε αναφερόμαστε ως χρονικά μεταβαλλόμενα συντομότερα μονοπάτια. Αξίζει να σημειώσουμε ότι η επίλυση του τελευταίου προβλήματος μπορεί να οδηγήσει και στην προσέγγιση ελάχιστων τιμών που περιγράφονται από άγνωστες συναρτήσεις καθώς εμπεριέχει έμμεσα τον υπολογισμό της συνάρτησης $A[P](t)$ για κάθε χρονική στιγμή t .

2.1.1 Κατά τμήματα γραμμικές συναρτήσεις και σημεία καμπής

Στο εξής θεωρούμε ότι οι συναρτήσεις μετάβασης και οι συναρτήσεις άφιξης που περιγράφονται στις ακμές είναι *κατά τμήματα γραμμικές* (piecewise linear) συναρτήσεις. Μια κατά-τμήματα συνάρτηση (χωρισμένη δηλαδή σε περιπτώσεις επιμέρους συναρτήσεων) καλείται γραμμική αν κάθε τμήμα της συνάρτησης είναι μια γραμμική συνάρτηση. Ο λόγος για τον οποίο θεωρούμε τέτοιες συναρτήσεις είναι ότι δύναται να προσεγγιστεί ο άγνωστος χρόνος μετάβασης από τέτοιου είδους συναρτήσεις: μπορούμε να εφαρμόσουμε μια περιοδική δειγματοληψία των χρόνων μετάβασης σε ακμές για συγκεκριμένες στιγμές αναχώρησης, και στη συνέχεια να θεωρήσουμε ως (συνεχή) συνάρτηση μετάβασης την κατά τμήματα γραμμική συνάρτηση που προκύπτει από την γραμμική παρεμβολή (interpolation) των σημείων δειγματοληψίας. Ακριβώς αυτή η διαδικασία εφαρμόζεται στην πράξη.

Συμβολίζουμε με K το συνολικό πλήθος γραμμικών τμημάτων που βρίσκονται σε όλες τις συναρτήσεις χρόνου άφιξης $A_e(t)$. Ως *σημείο καμπής* (breakpoint) εννοούμε την τιμή της χρονικής στιγμής t κατά την οποία η κλίση της συνάρτησης χρόνου άφιξης αλλάζει. Σε περιπτώσεις που θεωρούμε κατά τμήματα γραμμικές συναρτήσεις, το πλήθος των σημείων καμπής και το πλήθος των γραμμικών τμημάτων διαφέρουν μόνο κατά μια μονάδα, και επομένως για χάρη απλότητας θα χρησιμοποιούμε τις δυο αυτές τιμές εναλλάξ ανάλογα με το περιεχόμενο της ανάλυσης.

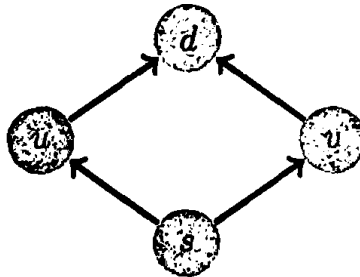
Θεωρούμε το πρόβλημα **ΧΡΟΝΙΚΑ ΜΕΤΑΒΑΛΛΟΜΕΝΟ ΣΥΝΤΟΜΟΤΕΡΟ ΜΟΝΟΠΑΤΙ** όπου κάθε συνάρτηση χρόνου άφιξης $A_e(t)$ περιγράφεται από μια κατά τμήματα γραμμική συνάρτηση. Τα σημεία καμπής στην συνάρτηση $A_e(t)$ καλούνται *πρωταρχικά σημεία καμπής* (primitive breakpoints). Η συνάρτηση $A[P](t)$ για κάποιο μονοπάτι P αποτελείται από την



σύνθεση επιμέρους συναρτήσεων χρόνου άφιξης στις ακμές. Τα σημεία καμπής της $A[P](t)$ που εμφανίζονται εξαιτίας των πρωταρχικών σημείων καμπής των επιμέρους συναρτήσεων ονομάζονται *πρωταρχικές εικόνες* (primitive images). Πιο συγκεκριμένα, μια πρωταρχική εικόνα b_F σε μια συνάρτηση $F = f_q(f_{q-1}(\dots f_1(t)))$ που εμφανίζεται για $t = t_F$ είναι η εικόνα ενός πρωταρχικού σημείου καμπής b σε κάποια συνάρτηση f_i αν η b εμφανίζεται για $t = t_i$ στην συνάρτηση $f_i(t)$ και $t_i = f_{i-1}(f_{i-2}(\dots f_1(t_F)))$.

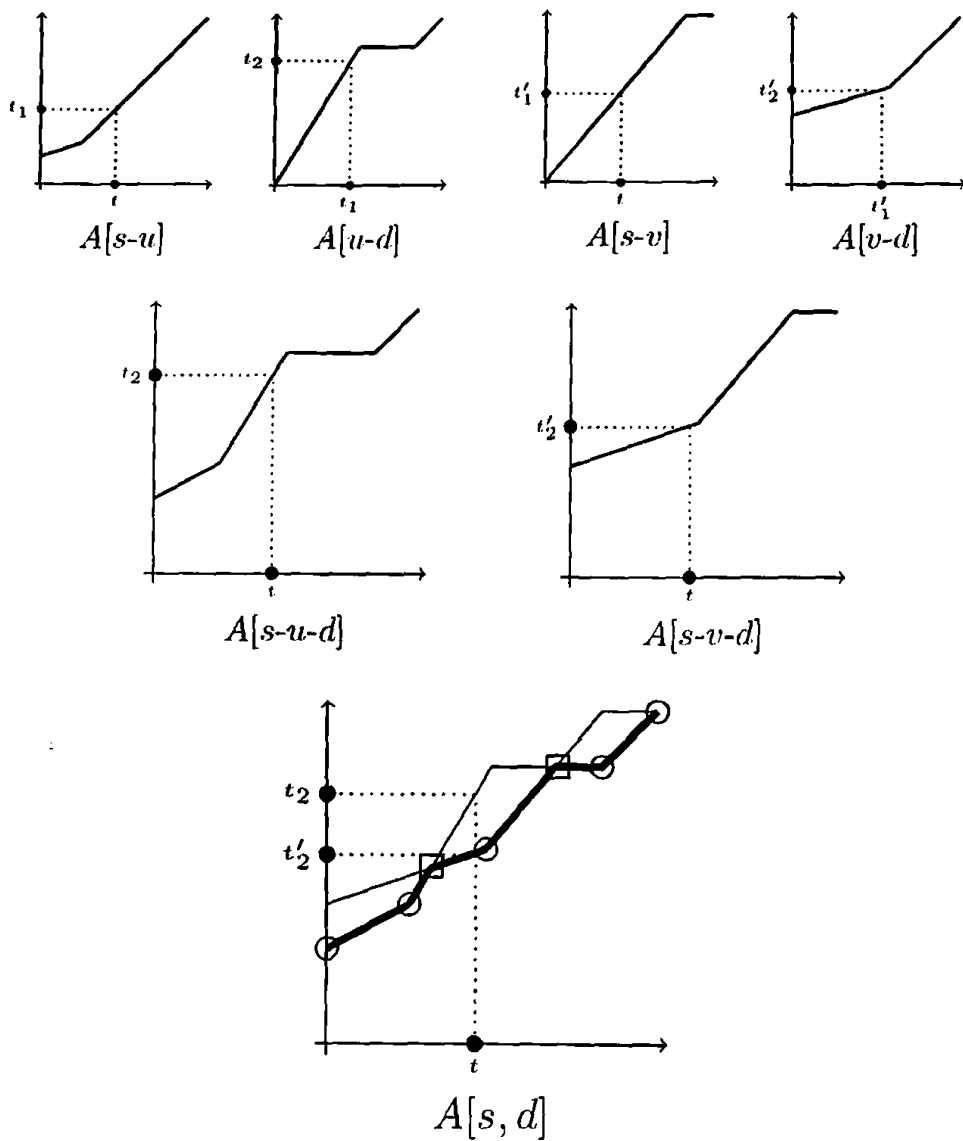
Τα σημεία καμπής της συνάρτησης $A[s, d]$ είναι δύο ειδών:

- *πρωτογενής καμπή* (primitive breakpoint), όπως περιγράφηκαν προηγουμένως και εμφανίζονται από την σύνθεση επιμέρους συναρτήσεων καθώς οι πρωταρχικές εικόνες του συντομότερου μονοπατιού μεταφέρονται και στην συνάρτηση $A[s, d]$, και
- *καμπή ελαχιστοποίησης* (minimization breakpoint): αναφέρονται σε εκείνα τα σημεία καμπής που εμφανίζονται στην συνάρτηση συντομότερης άφιξης $A[s, d](t)$ κάποια χρονική στιγμή t αν ο χρόνος άφιξης του μονοπατιού P_1 , που ήταν βέλτιστο την χρονική στιγμή $t - \epsilon$, γίνεται μεγαλύτερος από έναν άλλο χρόνο άφιξης κάποιου άλλου μονοπατιού P_2 την χρονική στιγμή $t + \epsilon$ για ένα απειροελάχιστο ϵ . Ένα ελάχιστο σημείο καμπής b εμφανίζεται εξαιτίας της ελαχιστοποίησης της συνάρτησης χρόνου σε κάποια συγκεκριμένη κορυφή v που βρίσκεται μετά την αφετηρία s μέχρι και να φτάσουμε στον προορισμό, δηλαδή στο κομμάτι $P_1 \cap P_2$.



Σχήμα 2.1: Ένα γράφημα G

Στο Σχήμα 2.2 δείχνουμε ένα παράδειγμα για το γράφημα του σχήματος 2.1. Στην πρώτη γραμμή δείχνουμε τις συναρτήσεις χρόνου άφιξης για κάθε ακμή του γραφήματος. Τα σχήματα της δεύτερης γραμμής αποτελούν τις συναρτήσεις χρόνου άφιξης των μονοπατιών $s - u - d$ και $s - v - d$, αντίστοιχα, που έχουν προκύψει από την σύνθεση των συναρτήσεων στις ακμές. Επίσης δείχνουμε την εξέλιξη του χρόνου: αν ακολουθήσουμε το μονοπάτι $s - u - d$ στον χρόνο αναχώρησης t θα φτάσουμε στην κορυφή u την χρονική στιγμή t_1 , και στη συνέχεια φτάνουμε στον προορισμό d την χρονική στιγμή t_2 . Όταν ακολουθούμε το μονοπάτι $s - v - d$ οι αντίστοιχες χρονικές στιγμές είναι t'_1 στην κορυφή v και t'_2 στην προορισμό d . Από τα δύο αυτά μονοπάτια, το συντομότερο είναι το δεύτερο για την χρονική στιγμή αναχώρησης t όπως φαίνεται στην συνάρτηση $A[s, d]$. Το σχήμα για την συνάρτηση $A[s, d]$ δείχνει επίσης τις πρωτογενείς καμπές (κύκλοι) και τις καμπές ελαχιστοποίησης (τετράγωνα). Άτυπα μπορούμε να εκφράσουμε τα ελάχιστα σημεία καμπής



Σχήμα 2.2: Οι συναρτήσεις χρόνου άφιξης για την κάθε ακμή του γραφήματος G του Σχήματος 2.1 και οι αντίστοιχες συνθέσεις των συναρτήσεων αυτών.

ως την λειτουργία ελαχιστοποίησης των συναρτήσεων χρόνου άφιξης μεταξύ όλων των μονοπατιών $P \in \mathcal{P}_{s,d}$, ενώ τα πρωταρχικά σημεία καμπής είναι εκείνα τα σημεία καμπής που εμφανίζονται εξαιτίας της συνάρτησης άφιξης στο κάθε μονοπάτι που προκύπτει από την συναρτησιακή σύνθεση των κατά τμήματα γραμμικών συναρτήσεων στις ακμές. Η συνάρτηση συντομότερης άφιξης $A[s, d]$ είναι (ως εφαρμογή του τελεστή $\min$ στις συναρτήσεις άφιξης των $s - u - d$ και $s - v - d$) ο κάτω-φάκελος των $A[s - u - d]$ και $A[s - v - d]$.

Κλείνουμε την ενότητα αυτή με ένα άνω όριο ως προς το πλήθος των πρωταρχικών σημείων καμπής.

Λήμμα 2.1 ([15]). Σε κάθε συνάρτηση συντομότερης άφιξης $A[s, d](t)$ υπάρχουν το πολύ K πρωταρχικές εικόνες όπου K είναι το πλήθος των γραμμικών τμημάτων (= σημεία καμπής) που περιγράφουν τις συναρτήσεις χρόνου άφιξης ακμών.



2.1.2 Ο περιορισμός σε FIFO δίκτυα

Το γενικό πρόβλημα εύρεσης χρονικά μεταβαλλόμενου συντομότερου μονοπατιού είναι τουλάχιστον NP-δύσκολο καθώς σε αυτό ανάγονται μια πληθώρα NP-δύσκολων προβλημάτων, όπως είναι το πρόβλημα των σακιδίων (knapsack problem) [31]. Βέβαια ανάλογα με τον τρόπο που ορίζει κανείς το πρόβλημα μπορεί να μην ανήκει ούτε στην κλάση NP διότι η έξοδος του προβλήματος μπορεί να μην φράσσεται πολυωνυμικά. Επιπλέον, όπως έδειξαν οι Orla και Rom [32], υπάρχουν στιγμιότυπα του προβλήματος με πεπερασμένο ελάχιστο χρόνο μετάβασης, όπου όμως όλα τα βέλτιστα (προφανώς όχι απλά) μονοπάτια απαρτίζονται από άπειρες εμφανίσεις ακμών. Αυτό ισχύει όταν δεν επιτρέπεται η αναμονή σε ενδιάμεσους κόμβους, διότι, διαφορετικά, αντί να κάνει κανείς έναν κύκλο που απλά προκαλεί μια καθυστέρηση, μπορεί απλά να περιμένει στον πρώτο κόμβο του κύκλου δίχως να κινείται μέχρι τη στιγμή που θα επέστρεφε από τον κύκλο και τότε να συνεχίσει δίχως να ακολουθεί τον κύκλο.

Όταν τα πακέτα σε ένα δίκτυο που ταξιδεύουν δια μέσου των ακμών σέβονται την FIFO ιδιότητα τότε η υπολογιστική πολυπλοκότητα του προβλήματος αλλάζει. Για κάθε ακμή (u, v) που σέβεται την FIFO ιδιότητα θα πρέπει να ισχύει για οποιεσδήποτε δύο χρονικές στιγμές t, t' :

$$t < t' \Rightarrow A_{(u,v)}(t) \leq A_{(u,v)}(t').$$

Πολλές φορές η FIFO ιδιότητα αναφέρεται και ως ιδιότητα *μη-προσπέρασης* διότι ουσιαστικά αναφέρει ότι αν ένα αντικείμενο k_1 αναχωρήσει από την κορυφή u την χρονική στιγμή t και στη συνέχεια ένα άλλο αντικείμενο k_2 αναχωρήσει από την κορυφή u την χρονική στιγμή $t' > t$ τότε το αντικείμενο k_2 δεν μπορεί να φτάσει στην κορυφή v νωρίτερα από το αντικείμενο k_1 χρησιμοποιώντας την ακμή (u, v) .

Στην πράξη πολλά δίκτυα, ειδικά δίκτυα μεταφορών, επιδεικνύουν FIFO συμπεριφορές. Παρόλο που τα FIFO δίκτυα είναι χρήσιμα για την μελέτη μέσων μεταφοράς που η προσπέραση είναι σπάνια (όπως σιδηροδρομικά δίκτυα), η μοντελοποίηση δικτύου αυτοκινητόδρομων επιφέρει δίκτυα που δεν ικανοποιούν απαραίτητα την FIFO ιδιότητα. Για το πρόβλημα του χρονικά μεταβαλλόμενου συντομότερου μονοπατιού η ύπαρξη της FIFO ιδιότητας είναι απαραίτητη για να καθορίσουμε ένα αποδεκτό επίπεδο υπολογισιμότητας:

- σε FIFO δίκτυα το πρόβλημα είναι πολυωνυμικά επιλύσιμο [25],
- ακόμα και με φωτεινούς σηματοδότες (στην ουσία δηλαδή με πολιτική και δυνατότητα αναμονής σε κάθε ενδιάμεση κορυφή) [2],
- ενώ παραμένει NP-δύσκολο σε δίκτυα χωρίς την FIFO ιδιότητα [31].

Σημειώνεται ότι μια ακμή που δεν παρουσιάζει την FIFO ιδιότητα μπορεί να προσομοιωθεί από μια FIFO ακμή αν επιτρέψουμε την αναμονή: κάποιος μπορεί να περιμένει στην ουρά της ακμής για να επιτύχει τη συντομότερη χρονική στιγμή άφιξης στην κεφαλή της ακμής. Δηλαδή κάθε non-FIFO δίκτυο με δυνατότητα αναμονής στους κόμβους δίχως περιορισμό ισοδυναμεί με (ανάγεται πολυωνυμικά σε) κάποιο FIFO δίκτυο (προφανώς) δίχως αναμονή στους κόμβους. Για τον λόγο αυτό θα επικεντρωθούμε σε μη-FIFO δίκτυα με

την δυνατότητα επιλογής αναμονής σε κορυφές του δικτύου για να αντιμετωπίσουμε ακόμα και καταστάσεις που δεν ικανοποιούν την FIFO ιδιότητα.

2.2 Επισκόπηση Βιβλιογραφίας για ΧΣΑ

Σε αυτή την ενότητα θα αναφερθούμε σε γνωστά αποτελέσματα για το πρόβλημα της εύρεσης χρονικά μεταβαλλόμενου συντομότερου μονοπατιού για συγκεκριμένη χρονική στιγμή αναχώρησης από την αφετηρία (ΧΡΟΝΟΙ ΣΥΝΤΟΜΟΤΕΡΗΣ ΑΦΙΞΗΣ, ΓΙΑ ΔΕΔΟΜΕΝΗ ΑΝΑΧΩΡΗΣΗ ΑΠΟ ΑΦΕΤΗΡΙΑ (ΧΣΑ)). Αρχίζουμε την μελέτη μας από τη στατική εκδοχή, και στη συνέχεια παρουσιάζουμε τις δημοφιλέστερες τεχνικές για χρονικά μεταβαλλόμενα δίκτυα.

2.2.1 Στατικοί αλγόριθμοι

Ας δούμε πρώτα τον κλασικό αλγόριθμο του Dijkstra για στατικά δίκτυα. Ο αλγόριθμος του Dijkstra επιλύει το πρόβλημα των συντομότερων μονοπατιών από την αφετηρία s προς κάθε άλλο προορισμό αν υπάρχουν μόνο μη-αρνητικά βάρη σε πουλωνυμικό χρόνο (στην ουσία, δηλαδή, επιλύει το πρόβλημα ΣΥΝΤΟΜΟΤΕΡΑ ΜΟΝΟΠΑΤΙΑ ΑΠΟ ΑΦΕΤΗΡΙΑ). Αναφέρεται συχνά ως μια μέθοδος ανάθεσης ετικετών (label-setting method) (επισκεπτόμενες κορυφές), σε αντιπαράθεση με τις label-correcting methods, όπως ο Bellman-Ford [14], και δουλεύει ως εξής: Για κάθε κορυφή v διατηρεί την εκτίμηση απόστασης $W_{s,v}$ από την αφετηρία s , τον προκάτοχο $p(v)$, και μια κατάσταση $S(v)$ που παίρνει τιμές από το σύνολο $\{\text{unreached, explored, settled}\}$. Αρχικά, $W_{s,v} = \infty$, $p(v) = \text{null}$ και $S(v) = \text{unreached}$ για κάθε κορυφή v . Η μέθοδος ξεκινάει θέτοντας $W_{s,s} = 0$, $p(s) = s$ και $S(s) = \text{explored}$. Όσο υπάρχουν επιγραφόμενες κορυφές (δηλαδή, $S(u) = \text{explored}$), η μέθοδος διαλέγει μια επισκεπτόμενη κορυφή u , εφαρμόζει χαλάρωση ακμής (edge relaxation) σε όλες τις εξερχόμενες ακμές από την u και θέτει $S(u) = \text{settled}$. Για να εφαρμόσει την διαδικασία relax στην ακμή (u, v) πρώτα ελέγχει αν $W_{s,v} > W_{s,u} + w(u, v)$ και σε περίπτωση που ισχύει θέτει $W_{s,v} = W_{s,u} + w(u, v)$, $p(v) = u$ και $S(v) = \text{explored}$. Ο Αλγόριθμος Στατικός-Dijkstra δίνεται σε ψευδοκώδικα στον Αλγόριθμο 1. Αν το γράφημα δεν περιέχει ακμές με αρνητικά βάρη, τότε ο αλγόριθμος του Dijkstra τερματίζει με τις σωστές συντομότερες αποστάσεις από την αφετηρία s [9]. Ο αλγόριθμος του Dijkstra εξετάζει τις κορυφές ως προς την τρέχουσα εκτίμηση $W_{s,v}$ κάθε επισκεφθείσας (και μη οριστικοποιημένης) κορυφής v από την αφετηρία s και κάθε κορυφή εξετάζεται το πολύ μια φορά. Η χρονική πολυπλοκότητα του αλγορίθμου εξαρτάται από την υλοποίηση της ουράς προτεραιότητας και όταν χρησιμοποιείται Fibonacci σωρός έχουμε τον ταχύτερο γνωστό χρόνο $O(m + n \log n)$ [13].

Αν ενδιαφερόμαστε μόνο για το συντομότερο μονοπάτι μεταξύ της αφετηρίας s και του προορισμού d τότε μπορούμε να τερματίσουμε νωρίτερα τον αλγόριθμο του Dijkstra αντί να εξετάσουμε όλες τις κορυφές του γραφήματος. Σε εκείνο το σημείο μπορούμε να χτίσουμε το συντομότερο (s, d) -μονοπάτι εξετάζοντας την συνάρτηση των προκατόχων $p(v)$ για κάθε κορυφή v . Ωστόσο αν αφήσουμε τον αλγόριθμο να υπολογίσει όλα τα συντομότερα



Αρχικοποίηση:

```
for all  $v \in V(G) \setminus \{s\}$  do
     $W_{s,v} \leftarrow \infty$ 
     $p(v) \leftarrow null$ 
     $S(v) \leftarrow \text{unreached}$ 
 $W_{s,s} \leftarrow 0$ 
 $p(s) \leftarrow s$ 
 $Q \leftarrow V(G)$  /* Use a Priority Queue  $Q$  */
 $S(s) \leftarrow \text{explored}$ 
```

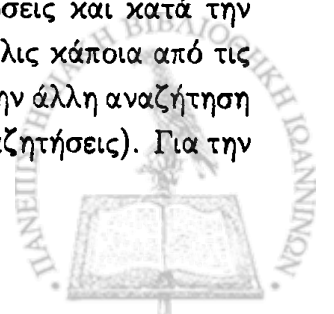
Βασικός βρόγχος:

```
while  $Q \neq \emptyset$  do
    Select  $u \in Q$  with the minimum  $W_{s,u}$  /* priority key is  $W_{s,u}$  */
     $Q \leftarrow Q \setminus \{u\}$ 
    for all  $(u, v) \in E(G)$  do
        if  $W_{s,v} > W_{s,u} + w(u, v)$  then /* Relaxing edge  $(u, v)$  */
             $W_{s,v} \leftarrow W_{s,u} + w(u, v)$ 
             $p(v) \leftarrow u$ 
             $S(v) \leftarrow \text{explored}$ 
         $S(u) \leftarrow \text{settled}$ 
```

Αλγόριθμος 1: Στατικός-Dijkstra

μονοπάτια από την αφετηρία s προς κάθε άλλη κορυφή τότε κατασκευάζουμε αυτό που ονομάζουμε δέντρο συντομότερων μονοπατιών.

Μια σημαντική παραλλαγή του Dijkstra αλγορίθμου για το συντομότερο (s, d) -μονοπάτι είναι η *αμφίδρομη αναζήτηση* (bidirectional search) [30]: αντί να χτίζουμε μόνο ένα δέντρο ελάχιστων μονοπατιών ξεκινώντας από την αφετηρία s , χτίζουμε επίσης με τον ίδιο τρόπο ένα δέντρο ελάχιστων μονοπατιών από τον προορισμό d στο ανεστραμμένο γράφημα $\overleftarrow{G}$. Κατά την αρχικοποίηση στην ευθεία αναζήτηση ελέγχουμε την αφετηρία s και στην προς-τα-πίσω αναζήτηση ελέγχουμε τον προορισμό d . Ο αλγόριθμος αποτελείται από δύο φάσεις. Στην πρώτη φάση ο αλγόριθμος διατηρεί το μήκος μ του συντομότερου μονοπατιού που έχει ανακαλύψει ως εκείνη την στιγμή και το αντίστοιχο συντομότερο μονοπάτι. Αρχικά θέτουμε $\mu = \infty$. Όταν εξετάζουμε μια ακμή (u, v) κατά την ευθεία αναζήτηση και η κορυφή v έχει ήδη ανακαλυφθεί (δηλαδή η κορυφή v έχει πάρει την επιγραφή *settled*) από την προς-τα-πίσω αναζήτηση τότε γνωρίζουμε τα μήκη των συντομότερων (s, u) -μονοπατιού και (d, v) -μονοπατιού με τιμές W_u^f και W_v^b , αντίστοιχα. Αν $\mu > W_u^f + W_v^b + w(u, v)$ τότε βρήκαμε ένα συντομότερο μονοπάτι και ενημερώνουμε κατάλληλα την ποσότητα μ και το αντίστοιχο συντομότερο μονοπάτι. Εφαρμόζουμε αντίστοιχες ενημερώσεις και κατά την προς-τα-πίσω αναζήτηση. Ο αλγόριθμος στην πρώτη φάση τερματίζει μόλις κάποια από τις δύο αναζητήσεις επιλέγει κάποια κορυφή που έχει ήδη ανακαλυφθεί από την άλλη αναζήτηση (δηλαδή, μόλις μια κορυφή k πάρει την επιγραφή *settled* και στις δύο αναζητήσεις). Για την

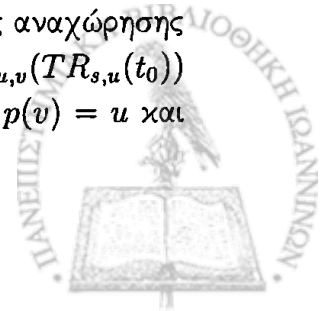


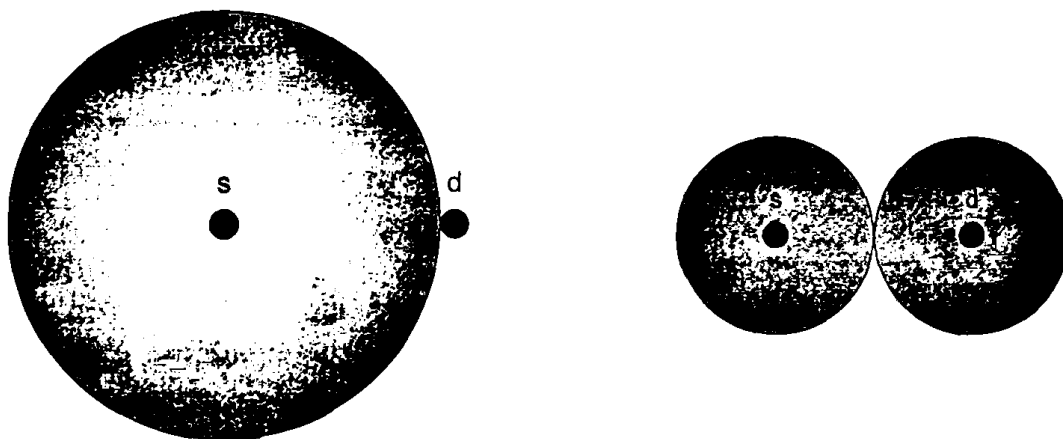
δεύτερη φάση του αλγορίθμου γνωρίζουμε το εξής: έστω $\vec{P}(u)$ το συντομότερο μονοπάτι από την αφετηρία s που έχτισε η ευθεία αναζήτηση για κάποια κορυφή u στην πρώτη φάση του αλγορίθμου και έστω $\overleftarrow{P}(v)$ το συντομότερο μονοπάτι από την κορυφή v προς τον προορισμό d που έχτισε η προς-τα-πίσω αναζήτηση στην πρώτη φάση του αλγορίθμου. Τότε το συντομότερο (s, d) -μονοπάτι είναι είτε το $\vec{P}(k) \cap \overleftarrow{P}(k)$ όπου k η κοινή κορυφή που ανακάλυψαν οι δύο αναζητήσεις, ή το μονοπάτι $\vec{P}(u) \cup (u, v) \cup \overleftarrow{P}(v)$ για κάποια ακμή (u, v) όπου u και v κορυφές που ανακάλυψαν οι δυο αναζητήσεις, αντίστοιχα, στην πρώτη φάση. Επομένως η δεύτερη φάση εξετάζει όλες τις πιθανές κορυφές με βάση τον παραπάνω ισχυρισμό και χτίζει το συντομότερο (s, d) -μονοπάτι. Σημειώνουμε ότι υπάρχει και άλλο κριτήριο τερματισμού για την αμφίδρομη αναζήτηση που δεν απαιτεί την εκτέλεση της δεύτερης φάσης που περιγράψαμε προηγουμένως. Το κριτήριο τερματισμού της πρώτης (και τελικής) φάσης αποτελεί ο έλεγχος $\text{top}^f + \text{top}^b \geq \mu$ όπου top^f είναι το μήκος του μονοπατιού από την πηγή s προς το πρώτο στοιχείο της ουράς προτεραιότητας της ευθείας αναζήτησης και top^b είναι το μήκος του μονοπατιού από την προορισμό d προς το πρώτο στοιχείο της ουράς προτεραιότητας της προς-τα-πίσω αναζήτησης. Μόλις ισχύει το κριτήριο τερματισμού τότε ο αλγόριθμος πράγματι τερματίζει με το σωστό συντομότερο μονοπάτι [17].

Επειδή μπορούμε να φανταστούμε τον αλγόριθμο του Dijkstra ως εξερεύνηση κορυφών σε κυκλική αναζήτηση με κέντρο την αφετηρία s με αυξανόμενη ακτίνα έως ότου ο προορισμός d βρεθεί, η αμφίδρομη αναζήτηση είναι ταχύτερη διότι ανακαλύπτει κορυφές σε δύο κύκλους με κέντρα τα s, d έως ότου οι δύο αυτοί κύκλοι ακουμπήσουν ο ένας τον άλλο (Σχήμα 2.3). Τότε ο εσωτερικός χώρος (ο χώρος αναζήτησης) μέσα στους δύο κύκλους που αναπαριστά το πλήθος των κορυφών που ανακαλύπτει θα είναι, ανάλογα με την τοπολογία του δικτύου, δύο φορές μικρότερος από την μονόδρομη περίπτωση. Αν και στην πράξη αυτό πολλές φορές ισχύει, μπορεί κανείς εύκολα να φτιάξει παραδείγματα όπου ο χώρος αναζήτησης του μονόδρομου Dijkstra είναι ακριβώς ο ίδιος με το χώρο αναζήτησης του αμφίδρομου Dijkstra. Αξίζει να σημειώσουμε ότι σε ένα χρονικά μεταβαλλόμενο περιβάλλον δεν μπορούμε να εφαρμόσουμε την αμφίδρομη διαδικασία καθώς είναι άγνωστος ο χρόνος άφιξης στον προορισμό.

2.2.2 Αλγόριθμοι για το πρόβλημα Χρόνου Συντομότερης Άφιξης

Σε χρονικά μεταβαλλόμενο περιβάλλον σε FIFO δίκτυα υπάρχει αλγόριθμος που βασίζεται στην επέκταση του Dijkstra αλγορίθμου σε στατικό περιβάλλον [11]. Θυμίζουμε ότι στην περίπτωση αυτή ο αλγόριθμος δέχεται ως είσοδο μια συνάρτηση χρόνου άφιξης $A_e(t)$ για κάθε ακμή του γραφήματος (σε αντικατάσταση των βαρών $w(e)$ της στατικής περίπτωσης) και την χρονική στιγμή εκκίνησης t_0 από την αφετηρία s . Επίσης για κάθε κορυφή v θα χρησιμοποιήσουμε μια εκτίμηση συντομότερης άφιξης στην v ως $TR_{s,v}(t_0)$ αντί για την εκτίμηση απόστασης $W_{s,v}$ που είχαμε στην κλασική περίπτωση. Η βασική διαφοροποίηση βρίσκεται στη διαδικασία relax μιας ακμής (u, v) : αν t_0 είναι ο χρόνος αναχώρησης από την αφετηρία τότε ελέγχουμε την συνθήκη $TR_{s,v}(t_0) > TR_{s,u}(t_0) + D_{u,v}(TR_{s,u}(t_0))$ και αν ικανοποιείται τότε θέτουμε $TR_{s,v}(t_0) = TR_{s,u}(t_0) + D_{u,v}(TR_{s,u}(t_0))$, $p(v) = u$ και





Σχήμα 2.3: Ο χώρος αναζήτησης του μονόδρομου στατικού Dijkstra και ο χώρος αναζήτησης του αμφίδρομου στατικού Dijkstra.

$S(v) = \text{explored}$. Θυμίζουμε ότι για μια συνάρτηση χρόνου άφιξης ακμής $A_{u,v}(t)$ ο χρόνος μετάβασης περιγράφεται από την συνάρτηση $D_{u,v}(t) = A_{u,v}(t) - t$. Η αποτελεσματικότητα του αλγορίθμου εξαρτάται από τον κανόνα που επιλέγουμε την επόμενη κορυφή προς εξέταση. Λέμε ότι η $TR_{s,u}(t_0)$ είναι *ακριβής* αν ισούται με την απόσταση από την s στην u όταν ο χρόνος αναχώρησης από την s είναι t_0 . Αν κάποιος διαλέγει πάντα μια κορυφή u τέτοια ώστε κατά την διάρκεια της επιλογής $TR_{s,u}(t_0)$ να είναι ακριβής τότε κάθε κορυφή εξετάζεται το πολύ μια φορά. Σε αυτή την περίπτωση χρειάζεται να εφαρμόσουμε την διαδικασία *relax* σε ακμές (u, v) όπου $S(u) \neq \text{settled}$ και ο αλγόριθμος καλείται *ανάθεσης-ετικετών* (label-setting). Ο κλασικός αλγόριθμος του Dijkstra είναι ένας από τους αλγορίθμους ανάθεσης-ετικετών για συντομότερα μονοπάτια. Αυτό σημαίνει ότι κάθε ετικέτα οριστικοποιείται μια φορά και δεν τροποποιείται (μέσω χαλάρωσης κάποιας εισερχόμενης ακμής) έπειτα από την μονιμοποίησή της.

Η χρονική πολυπλοκότητα του αλγορίθμου εξαρτάται (α) όπως και στην περίπτωση του στατικού Dijkstra από την υλοποίηση της ουράς προτεραιότητας που η πιο γρήγορη υλοποίηση γίνεται σε $O(m + n \log n)$ χρόνο [31, 32] και (β) από το μέγιστο πλήθος των σημείων καμπής των συναρτήσεων χρόνου άφιξης όταν αυτές περιγράφονται από κατά-τμήματα γραμμικές συναρτήσεις. Δηλαδή η συνολική πολυπλοκότητα χρόνου θα είναι $O(m \log \log K_{\max} + n \log n)$ όπου $K_{\max}$ είναι το μέγιστο πλήθος των σημείων καμπής από όλες τις συναρτήσεις χρόνου άφιξης των ακμών. Ο Αλγόριθμος TD-Dijkstra δίνεται σε ψευδοκώδικα στον Αλγόριθμο 2.

Ο αλγόριθμος TD-Dijkstra υπολογίζει βαθμιαία τις τιμές των συντομότερων αφίξεων $TR_{s,v}(t)$ για μια δεδομένη χρονική στιγμή t . Στη συνέχεια παρουσιάζουμε τον αλγόριθμο TD-Bellman-Ford διορθώνοντας-επιγραφές (label correcting) που υπολογίζει επιπλέον τις τιμές των συναρτήσεων $TR_{s,v}$ σε οποιαδήποτε χρονική στιγμή [31, 32]. Η ονομασία διορθώνοντας-επιγραφές υπονοεί ότι η επιγραφή μιας κορυφής ($TR_{s,v}$) δεν είναι καθορισμένη ακόμα και όταν η κορυφή εξάγεται από την ουρά προτεραιότητας καθώς μπορεί να επανατοποθετηθεί στην ουρά πολλαπλές φορές σε αντίθεση με τον κλασικό Dijkstra αλγόριθμο. Περιγράφουμε έναν τέτοιο αλγόριθμο που βασίζεται στις εργασίες [6, 31, 32]. Η

Αρχικοποίηση:

```
for all  $v \in V(G) \setminus \{s\}$  do
   $TR_{s,v}(t_0) \leftarrow \infty$ 
   $p(v) \leftarrow null$ 
   $S(v) \leftarrow unreached$ 
 $TR_{s,s}(t_0) \leftarrow t_0$ 
 $p(s) \leftarrow s$ 
 $Q \leftarrow V(G)$  /* Use a Priority Queue  $Q$  */
 $S(s) \leftarrow explored$ 
```

Βασικός βρόγχος:

```
while  $Q \neq \emptyset$  do
  Select  $u \in Q$  with the minimum  $TR_{s,u}(t_0)$  /* priority key is  $TR_{s,u}(t_0)$  */
   $Q \leftarrow Q \setminus \{u\}$ 
  for all  $(u, v) \in E(G)$  such that  $S(v) \neq settled$  do
    if  $TR_{s,v}(t_0) > TR_{s,u}(t_0) + D_{u,v}(TR_{s,u}(t_0))$  then /* Relaxing edge  $(u, v)$  */
       $TR_{s,v}(t_0) \leftarrow TR_{s,u}(t_0) + D_{u,v}(TR_{s,u}(t_0))$ 
       $p(v) \leftarrow u$ 
       $S(v) \leftarrow explored$ 
   $S(u) \leftarrow settled$ 
```

Αλγόριθμος 2: TD-Dijkstra

επιγραφή $TR_{s,v}$ μιας κορυφής v υπολογίζεται με βαθμιαίο τρόπο για την κλασική περίπτωση του Dijkstra, ενώ τώρα η επιγραφή θα είναι μια συνάρτηση του χρόνου. Πιο συγκεκριμένα κατά τον τερματισμό του αλγορίθμου θέλουμε να ισχύει $TR_{s,v} = TR_{s,v}(t)$. Θεωρούμε $\mathcal{T}$ το διάστημα του χρόνου που μας ενδιαφέρει, δηλαδή $\mathcal{T} \subset [0, \infty)$. Κατά την αρχικοποίηση του αλγορίθμου θέτουμε σταθερές συναρτήσεις ως επιγραφές: $\forall t \in \mathcal{T}, TR_{s,s}(t) = t$ και $TR_{s,v}(t) = \infty, \forall v \in V(G) \setminus \{s\}$.

Στον TD-Bellman-Ford αλγόριθμο, εξετάζουμε σε κάθε γύρο όλες τις ακμές μήπως κάποια από αυτές ικανοποιεί το κριτήριο χαλάρωσης (relaxation). Σε κάθε επανάληψη διαλέγουμε κάποια κορυφή από την ουρά και δοκιμάζουμε να κάνουμε χαλάρωση των εξερχόμενων ακμών της. Η κορυφή προσωρινά εξέρχεται από την ουρά, αλλά ενδέχεται στο μέλλον να επανεισέλθει σε αυτήν, εξαιτίας χαλάρωσης κάποιας εισερχόμενης σε αυτήν ακμής. Πιο συγκεκριμένα σε κάθε επανάληψη εξάγουμε μια κορυφή u από την ουρά προτεραιότητας με την ελάχιστη $TR_{s,u}$ (δηλαδή, $\min_{t \in \mathcal{T}} TR_{s,u}(t)$) και εφαρμόζουμε την διαδικασία relax στις εξερχόμενες ακμές της u : για κάθε ακμή $(u, v) \in E(G)$ μια προσωρινή επιγραφή $k_v(t) = TR_{s,u}(t) + D_{u,v}(TR_{s,u}(t))$ δημιουργείται. Τότε ελέγχουμε αν $k_v(t) < TR_{s,v}(t)$ για κάποια χρονική στιγμή $t \in \mathcal{T}$ και αν ισχύει τότε η ακμή (u, v) δίνει μια βελτίωση για τουλάχιστον μια χρονική στιγμή του διαστήματος $\mathcal{T}$. Επομένως ενημερώνουμε την επιγραφή ως εξής: $TR_{s,v}(t) = \min\{TR_{s,v}(t), k_v(t)\}$. Ο αλγόριθμος σταματάει την εκτέλεσή του μόλις εξαχθεί κάποια κορυφή u από την ουρά προτεραιότη-



τας για την οποία $\min_{t \in \mathcal{T}} TR_{s,u}(t) \geq \max_{t \in \mathcal{T}} TR_{s,d}(t)$. Ο Αλγόριθμος TD-Bellman-Ford (διορθώνοντας-επιγραφές, label-correcting) δίνεται σε ψευδοκώδικα στον Αλγόριθμο 3.

Αρχικοποίηση:

```

for all  $v \in V(G) \setminus \{s\}$  do
   $TR_{s,v}(t) \leftarrow \infty, \forall t \in \mathcal{T}$ 
   $p(v) \leftarrow null$ 
   $S(v) \leftarrow unreached$ 
 $TR_{s,s}(t) \leftarrow t, \forall t \in \mathcal{T}$ 
 $p(s) \leftarrow s$ 
 $Q \leftarrow V(G)$  /* Use a Priority Queue  $Q$  */
 $S(s) \leftarrow explored$ 

```

Βασικός βρόγχος:

```

while  $Q \neq \emptyset$  do
  Select  $u \in Q$  with the minimum  $\min_{t \in \mathcal{T}} TR_{s,u}(t)$  /* priority key is  $\min_{t \in \mathcal{T}} TR_{s,u}(t)$  */
   $Q \leftarrow Q \setminus \{u\}$ 
  if  $\min_{t \in \mathcal{T}} TR_{s,u}(t) \geq \max_{t \in \mathcal{T}} TR_{s,d}(t)$  then
    break
  for all  $(u, v) \in E(G)$  such that  $S(v) \neq settled$  do
     $k_v(t) \leftarrow TR_{s,u}(t) + D_{u,v}(TR_{s,u}(t)), \forall t \in \mathcal{T}$ 
    if  $\exists t \in \mathcal{T} : k_v(t) < TR_{s,v}(t)$  then /* Relaxing edge  $(u, v)$  */
       $TR_{s,v}(t) \leftarrow k_v(t)$ 
       $p(v) \leftarrow u$ 
       $S(v) \leftarrow explored$ 
       $Q \leftarrow Q \cup \{v\}$ 
     $S(u) \leftarrow settled$ 

```

Αλγόριθμος 3: TD-Bellman-Ford (label-correcting)

Ο αλγόριθμος TD-Bellman-Ford διορθώνοντας-επιγραφές (label-correcting) μπορεί επομένως να υπολογίσει τα συντομότερα μονοπάτια από την αφετηρία s που αναχωρούν σε οποιαδήποτε χρονική στιγμή. Μπορεί κάποιος να αντιληφθεί τον αλγόριθμο αυτόν ως την επίλυση ενός συγκεκριμένου στιγμιότυπου για τον υπολογισμό της συνάρτησης $TR_{s,v}(t)$ όπως προηγουμένως μόνο που τώρα εφαρμόζει την διαδικασία για κάθε πιθανή τιμή για το t . Σε διακριτό χρόνο υπάρχει ο συγκεκριμένος αλγόριθμος προτάθηκε για την επίλυση του προβλήματος σε χρόνο $\mathcal{O}(nm|T|)$ [38]. Σε συνεχή χρόνο οι Orda και Rom πρότειναν στην ουσία τον ίδιο αλγόριθμο [31, 32]. Χωρίς την FIFO ιδιότητα κατάφεραν μόνο να αποδείξουν ότι ο αλγόριθμος κάποια στιγμή θα τερματίσει. Σε FIFO δίκτυα, ωστόσο, έδειξαν χρονική πολυπλοκότητα $\mathcal{O}(nmF)$, όπου F είναι ο χρόνος που χρειάζεται για να εκτελεστούν βασικές πράξεις (π.χ., άθροισμα, σύγκριση, ελάχιστο) σε συναρτήσεις όπου το πεδίο ορισμού (χρόνος αναχώρησης) είναι συνεχές. Στην περίπτωση των κατά τμήματα γραμμικών συναρτήσεων (piecewise linear functions), F είναι το πλήθος των τμημάτων

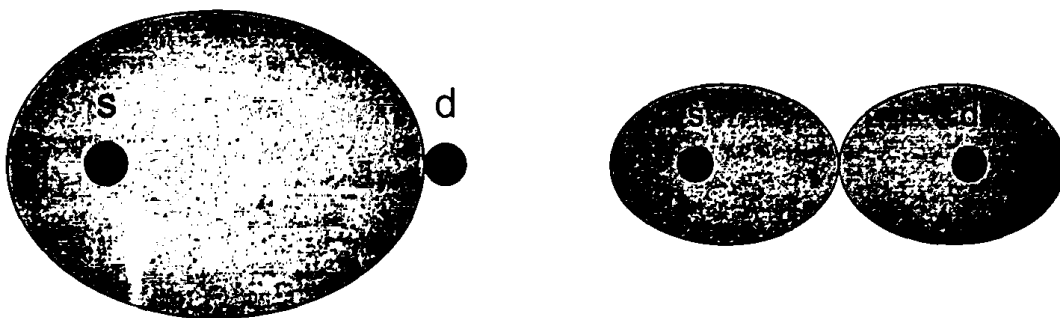
που μπορεί να εμφανιστούν στην συνάρτηση εξόδου $TR_{s,v}(t)$. Επομένως στο ερώτημα αν ο αλγόριθμος τρέχει σε πολυωνυμικό χρόνο η απάντηση εξαρτάται από την ύπαρξη υπερπολυωνυμικού (superpolynomial) πλήθους τμημάτων στην έξοδο $TR_{s,v}(t)$. Η απάντηση στο ερώτημα αυτό ήρθε το 2011, όπου οι Foschini, Hershberger, και Suri έδειξαν ότι το πλήθος αυτό είναι $n^{\Theta(\log n)}$ στην χειρότερη περίπτωση [15].

Στη συνέχεια θα περιγράψουμε δύο ακόμα αλγορίθμους που αποτελούν παραλλαγές του κλασικού Dijkstra αλγορίθμου και δουλεύουν τόσο σε στατικό όσο και χρονικά μεταβαλλόμενο περιβάλλον. Ο αλγόριθμος A^* είναι ο ένας από αυτούς και συχνά καλείται *στοχοπροσανατολισμένος* (goal-directed) αλγόριθμος. Η βασική διαφορά από τον κλασικό Dijkstra βρίσκεται στο κλειδί προτεραιότητας (priority key) όταν εξάγεται μια κορυφή από την ουρά. Στον αλγόριθμο A^* το κλειδί προτεραιότητας για μια κορυφή v αποτελείται από δύο κομμάτια: το μήκος του προσωρινού συντομότερου μονοπατιού από την αφετηρία προς την v (όπως και στον αλγόριθμο του Dijkstra), και το δεύτερο κομμάτι αποτελείται από μια υποεκτίμηση της απόστασης για να φτάσει στον προορισμό από την κορυφή v . Επομένως το κλειδί της v αναπαριστά μια εκτίμηση του μήκους του συντομότερου μονοπατιού από την αφετηρία s στον προορισμό d περνώντας δια μέσου της v και οι κορυφές ταξινομούνται στην ουρά με βάση αυτό το κριτήριο.

Η συνάρτηση εκτίμησης της απόστασης μεταξύ μιας κορυφής και του προορισμού καλείται *συνάρτηση δυναμικού* (potential function) π . Η χρησιμοποίηση της π έχει ως αποτέλεσμα να δίνει προτεραιότητα σε κορυφές που (υποτιθέμενα) είναι πιο κοντά στον προορισμό d . Αν η συνάρτηση δυναμικού είναι τέτοια ώστε $\pi(v) \leq \ell(v, d)$, για κάθε $v \in V(G)$, όπου $\ell(v, d)$ είναι η απόσταση από την v στον d , τότε ο αλγόριθμος A^* πάντα βρίσκει συντομότερα μονοπάτια [20]. Ο αλγόριθμος A^* συμπίπτει με τον Dijkstra σε ένα γράφημα όπου τα βάρη των ακμών είναι τα μειωμένα κόστη $w_\pi(u, v) = w(u, v) - \pi(u) + \pi(v)$ [22]. Από το γεγονός αυτό, μπορεί εύκολα κανείς να καταλάβει ότι αν $\pi(v) = 0, \forall v \in V(G)$ τότε ο A^* ανακαλύπτει τις ίδιες κορυφές με τον Dijkstra, ενώ αν $\pi(v) = \ell(v, d), \forall v \in V(G)$ τότε μόνο κορυφές στο συντομότερο μονοπάτι γίνονται διευθετημένες (settled), καθώς οι ακμές στο συντομότερο μονοπάτι έχουν μηδενικό μειωμένο κόστος. Δηλαδή ο αλγόριθμος A^* εξασφαλίζει ότι δεν πρόκειται να επισκεφτεί κορυφές που δεν επισκέπτεται ο Dijkstra. Πιο συγκεκριμένα, εάν $\pi(v)$ είναι μια καλή κάτω προσέγγιση της απόστασης για τον προορισμό τότε ο A^* οδηγεί με αποτελεσματικό τρόπο την αναζήτηση προς στον προορισμό. Δηλαδή ο χώρος αναζήτησης δεν είναι πλέον κυκλικός με κέντρο την αφετηρία s αλλά είναι ελλειπτικός με κατεύθυνση προς τον προορισμό d (Σχήμα 2.4). Ο αλγόριθμος A^* μπορεί εύκολα να εφαρμοστεί σε χρονικά μεταβαλλόμενο γράφημα με την FIFO ιδιότητα που θα τον ονομάζουμε TD- A^* , όπως δείξαμε και στις δύο περιπτώσεις του στατικού Dijkstra, αρκεί η συνάρτηση δυναμικού $\pi(v)$ να είναι ένα έγκυρο κάτω φράγμα των αποστάσεων $\ell(v, d)(t)$ για κάθε $t \in \mathcal{T}$ [4].

Σε ένα δίκτυο από αυτοκινητόδρομους οι Ευκλείδειες αποστάσεις μπορούν να χρησιμοποιηθούν για τον υπολογισμό των συναρτήσεων δυναμικού, πιθανώς διαιρεμένες με την μέγιστη επιτρεπτή ταχύτητα αν τα βάρη στις ακμές είναι χρόνοι ταξιδιού αντί για αποστάσεις. Αυτό προφανώς εφαρμόζεται και στην περίπτωση του χρονικά μεταβαλλόμενου





Σχήμα 2.4: Ο χώρος αναζήτησης του μονόδρομου TD-A* και ο χώρος αναζήτησης του αμφίδρομου TD-ALT.

περιβάλλοντος. Σε στατικά γραφήματα μια σημαντική βελτίωση πάνω σε Ευκλείδειες δυναμικές μπορεί να επιτευχθεί με την χρήση *κεντρικών σημείων* (landmarks). Η βασική ιδέα είναι να επιλέξουμε ένα μικρό σύνολο κορυφών του γραφήματος, επαρκώς εξαπλωμένα στο δίκτυο, και μια προεπεξεργασία να υπολογίσουμε όλες τις αποστάσεις μεταξύ αυτών των κορυφών (που καλούνται *κεντρικά σημεία*) και οποιασδήποτε κορυφής του συνόλου κορυφών. Τότε εκμεταλλευόμενοι την τριγωνική ανισότητα μπορούμε να εξάγουμε κάτω φράγματα για τις αποστάσεις μεταξύ δύο οποιονδήποτε κορυφών.

Ας υποθέσουμε ότι έχουμε διαλέξει ένα σύνολο $C \subset V(G)$ από κεντρικά σημεία και έχουμε αποθηκεύσει όλες τις αποστάσεις $\ell(u, c)$ και $\ell(c, v)$ για κάθε $v \in V(G) \setminus C$ και για κάθε $c \in C$. Τότε οι ακόλουθες τριγωνικές ανισότητες θα ισχύουν: $\ell(u, d) + \ell(d, c) \geq \ell(u, c)$ και $\ell(c, u) + \ell(u, d) \geq \ell(c, d)$. Επομένως η συνάρτηση $\pi_f(u) = \max_{c \in C} \{\ell(u, c) - \ell(d, c), \ell(c, d) - \ell(c, u)\}$ είναι ένα κάτω φράγμα της απόστασης $\ell(u, d)$ και μπορεί να χρησιμοποιηθεί ως μια έγκυρη συνάρτηση δυναμικού για την ευθεία αναζήτηση [17].

Σε αυτή την περίπτωση μπορεί να εφαρμοστεί αμφίδρομη αναζήτηση: μια εμπρόσθια αναζήτηση εφαρμόζεται στο G ξεκινώντας από την αφετηρία και χρησιμοποιώντας την συνάρτηση δυναμικού $\pi_f(u)$ που εκτιμάει την απόσταση για να φτάσει στον προορισμό και μια προς-τα-πίσω αναζήτηση εφαρμόζεται στο ανεστραμμένο γράφημα $\overleftarrow{G}$ από τον προορισμό χρησιμοποιώντας μια συνάρτηση δυναμικού $\pi_b(u)$ που εκτιμάει την απόσταση για να φτάσει στην αφετηρία. Η αμφίδρομη αναζήτηση με τις παραπάνω συναρτήσεις δυναμικού αποτελεί τον αλγόριθμο ALT. Είναι εύκολο να παρατηρήσει κανείς, ότι αν τα βάρη στις ακμές μπορούν μόνο να αυξάνουν ως προς την αρχική τιμή για τον υπολογισμό προς και από τα κεντρικά σημεία, τότε οι συναρτήσεις δυναμικού που είναι σχετικές με τα κεντρικά σημεία δίνουν έγκυρα κάτω φράγματα ακόμα και σε χρονικά μεταβαλλόμενα δίκτυα. Η τελευταία περίπτωση αποτελεί τον αλγόριθμο TD-ALT [8].

Υπάρχουν αρκετές ακόμα τεχνικές για τον υπολογισμό συντομότερων μονοπατιών για συγκεκριμένες χρονικές στιγμές αναχώρησης από την αφετηρία, τόσο σε στατικό όσο και σε χρονικά μεταβαλλόμενο περιβάλλον. Ο αναγνώστης που ενδιαφέρεται μπορεί να ανατρέξει στην πρόσφατη διδακτορική διατριβή του Nannicini [28] όπως επίσης και στις εργασίες [6, 7, 15, 16, 31, 32].



2.2.3 Αλγόριθμοι για το πρόβλημα Συναρτήσεων Συντομότερης Άφιξης

Όσο αφορά το πρόβλημα ΣΥΝΑΡΤΗΣΕΙΣ ΣΥΝΤΟΜΟΤΕΡΗΣ ΑΦΙΞΗΣ, ΓΙΑ ΑΝΑΧΩΡΗΣΗ ΑΠΟ ΔΕΔΟΜΕΝΗ ΑΦΕΤΗΡΙΑ (ΣΣΑ) οι αλγόριθμοι που επιλύουν τον υπολογισμό των συναρτήσεων χωρίζονται στις εξής κατηγορίες με βάση την δειγματοληψία ή την υπόθεση του χρόνου:

- (α) Το πεδίο των χρόνων αναχώρησης από την αφετηρία είναι διακριτό.
- (β) Το πεδίο των χρόνων αναχώρησης από την αφετηρία είναι συνεχές.

Για την περίπτωση (α) μπορεί κανείς να εφαρμόσει τους αλγόριθμους TD-Dijkstra ή TD-Bellman-Ford που περιγράψαμε προηγουμένως σε κάθε χρονική στιγμή που βρίσκεται στο εύρος του χρόνου, δηλαδή $\forall t \in T$. Επομένως διασπάμε το πρόβλημα του υπολογισμού των συναρτήσεων για διακριτό χρόνο σε $|T| + 1$ ενδιάμεσους υπολογισμούς. Η μέθοδος αυτή είναι ασφαλής καθώς οι συνθήκες βελτιστότητας δεν εξαρτώνται μεταξύ διαφορετικών χρονικών στιγμών για τον υπολογισμό της συνάρτησης $A[s, d](t)$ [6]. Η χρονική πολυπλοκότητα θα είναι $O(|T|(m + n \log n))$.

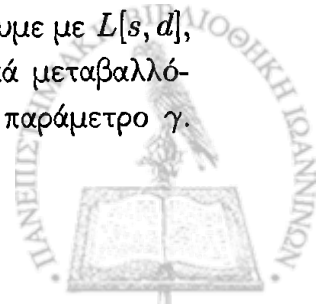
Για την περίπτωση (β) όπου το πεδίο των χρόνων αναχώρησης από την αφετηρία είναι συνεχές, υπάρχουν δύο αλγόριθμοι που επιλύουν τον υπολογισμό της $A[s, d](t)$ όταν οι συναρτήσεις άφιξης είναι κατά τμήματα γραμμικές [15]. Η χρονική πολυπλοκότητα του ενός αλγορίθμου εξαρτάται από την έξοδο και ο άλλος αλγόριθμος προσεγγίζει την συνάρτηση $A[s, d](t)$ με έναν σταθερό λόγο προσέγγισης. Θα περιγράψουμε αναλυτικά τους συγκεκριμένους αλγόριθμους για την περίπτωση αυτή στο Κεφάλαιο 3.

2.3 Παραμετροποιημένα Συντομότερα Μονοπάτια: Ένα Σχετικό Πρόβλημα

Στην παρούσα διατριβή μας απασχολεί ο υπολογισμός των συναρτήσεων συντομότερης άφιξης (earliest-arrival-functions), για οποιαδήποτε χρονική στιγμή αναχώρησης από δεδομένη αφετηρία. Εστιάζουμε το ενδιαφέρον μας σε κατά-τμήματα γραμμικές συναρτήσεις μετακίνησης (travel-time functions) για τις ακμές. Για το πρόβλημα αυτό έχει αποδειχτεί μια συσχέτιση με το πρόβλημα των παραμετροποιημένων συντομότερων μονοπατιών (parametric shortest paths) που θα παρουσιάσουμε στην ενότητα αυτή.

Στο πρόβλημα ΠΑΡΑΜΕΤΡΟΠΟΙΗΜΕΝΑ ΣΥΝΤΟΜΟΤΕΡΑ ΜΟΝΟΠΑΤΙΑ κάθε ακμή e του γραφήματος έχει κόστος $c_e(\gamma)$ που μεταβάλλεται γραμμικά ως προς μια παράμετρο γ , δηλαδή $c_e(\gamma) = a_e \cdot \gamma + b_e$. Το συντομότερο μονοπάτι από την αφετηρία s προς τον προορισμό d εξαρτάται από την τιμή της παραμέτρου γ και ο στόχος του προβλήματος είναι η εύρεση των συντομότερων μονοπατιών για κάθε τιμή της παραμέτρου γ .

Το μήκος του συντομότερου μονοπατιού μεταξύ του s και d το συμβολίζουμε με $L[s, d]$, σε αντιστοιχία με την συνάρτηση συντομότερης άφιξης $A[s, d](t)$ σε χρονικά μεταβαλλόμενο περιβάλλον, και το $L[s, d]$ είναι μια συνάρτηση που εξαρτάται από την παράμετρο γ .



Ορίζουμε ως $L[s, d] = \min(L[P] | P \in \mathcal{P}_{s,d})$ το ελάχιστο των γραμμικών συναρτήσεων μεταξύ (s, d) -μονοπατιών $P \in \mathcal{P}_{s,d}$. Θεωρούμε ως $B(L[s, d])$ το πλήθος των σημείων καμπής στο $L[s, d]$, που είναι δηλαδή το πλήθος των αλλαγών του συντομότερου μονοπατιού από την αφετηρία s προς των προορισμό d στο χώρο της παραμέτρου γ .

Μια σημαντική διαφορά μεταξύ των δύο μοντέλων (παραμετροποιημένα συντομότερα μονοπάτια και χρονικά μεταβαλλόμενα συντομότερα μονοπάτια) είναι η εξής που βασίζεται στην συνάρτηση μετάβασης: για τα παραμετροποιημένα συντομότερα μονοπάτια η συνάρτηση μετάβασης είναι απλά το άθροισμα των επιμέρους συναρτήσεων μετάβασης στις ακμές που απαρτίζουν το μονοπάτι, ενώ για τα χρονικά μεταβαλλόμενα συντομότερα μονοπάτια η συνάρτηση μετάβασης είναι η σύνθεση των επιμέρους συναρτήσεων μετάβασης στις ακμές που απαρτίζουν το μονοπάτι και αφαιρώντας το χρόνο αναχώρησης. Για να μπορέσει κανείς να συσχετίσει τα δύο αυτά μοντέλα θα πρέπει να αντιστοιχίσει μια συγκεκριμένη τιμή γ σε ένα εύρος χρονικών παραμέτρων χωρίς να μεταβάλει τα σημεία καμπής αρκετά και να κλιμακώσει προς τα κάτω τις καθυστερήσεις στις ακμές έτσι ώστε η παράμετρος $\gamma (= t)$ να μην αλλάζει σε μεγάλο βαθμό κατά την διάρκεια που ταξιδεύει κανείς από την αφετηρία s προς τον προορισμό d .

Το ακόλουθο αποτέλεσμα αναφέρει το πόσο πολύ μπορεί να αλλάξει το πλήθος των σημείων καμπής σε ένα παραμετροποιημένο συντομότερο μονοπάτι με γραμμικές συναρτήσεις ως προς την παράμετρο. Στην χειρότερη περίπτωση το συντομότερο μονοπάτι μεταξύ του s και d μπορεί να μεταβληθεί $n^{\Omega(\log n)}$ φορές καθώς η παράμετρος γ μεταβάλλεται και το αποτέλεσμα αυτό αποτελεί ένα κάτω φράγμα ως προς το πλήθος των σημείων καμπής (σημεία μεταβολής του συντομότερου μονοπατιού).

Θεώρημα 2.1 ([27]). *Υπάρχει γράφημα με γραμμικά παραμετροποιημένα βάρη στις ακμές και δύο κορυφές s και d τέτοιο ώστε το πλήθος των σημείων καμπής του $L(s, d)$ είναι $B(L[s, d]) = n^{\Omega(\log n)}$.*

Από την άλλη ένα άνω φράγμα είναι γνωστό από τον Gusfield για το πλήθος των σημείων καμπής που φράσσονται από $n^{O(\log n)}$ [19].

Επιπρόσθετα υπάρχουν και άλλα μοντέλα παραμετροποιημένων συντομότερων μονοπατιών που ανάλογα με τον περιορισμό στις συναρτήσεις κόστους κάθε ακμής οδηγούνται και σε ανάλογα αποτελέσματα. Για παράδειγμα, αν τα βάρη των ακμών ακολουθούν την συνάρτηση $c(e_i) = w(e_i) - \gamma$, όπου $w(e_i)$ το (σταθερό και στατικό) βάρος κάθε ακμής e_i τότε για το μοντέλο αυτό υπάρχουν μόνο πολυωνυμικό σε πλήθος διαφορετικά συντομότερα μονοπάτια και επιδέχεται αποτελεσματικούς αλγόριθμους [24, 37].

2.4 Φράγματα Πολυπλοκότητας Συναρτήσεων Συντομότερης Άφιξης

Στην ενότητα αυτή θα παρουσιάσουμε άνω και κάτω φράγματα για συναρτήσεις συντομότερης άφιξης σε χρονικά μεταβαλλόμενα δίκτυα. Μας ενδιαφέρει το πρόβλημα ΣΥΝΑΡΤΗΣΕΙΣ ΣΥΝΤΟΜΟΤΕΡΗΣ ΑΦΙΞΗΣ, ΓΙΑ ΑΝΑΧΩΡΗΣΗ ΑΠΟ ΔΕΔΟΜΕΝΗ ΑΦΕΤΗΡΙΑ (ΣΣΑ) (σε οποιαδήποτε χρονική στιγμή) όπου κάθε συνάρτηση χρόνου άφιξης περιγράφεται από

μια κατά τμήματα γραμμική συνάρτηση. Ειδικότερα δείχνουμε ότι σε ένα γράφημα με κατά τμήματα γραμμικές συναρτήσεις που περιγράφουν τα βάρη των ακμών, η συνάρτηση χρόνου άφιξης έχει πολυπλοκότητα $n^{\Theta(\log n)}$ στη χειρότερη περίπτωση. Πιο συγκεκριμένα, ακόμα και με ακμές με γραμμικά βάρη υπάρχει γράφημα για το οποίο το συντομότερο μονοπάτι από την αφετηρία s προς τον προορισμό d αλλάζει $n^{\Omega(\log n)}$ φορές ή ισοδύναμα η συνάρτηση χρόνου άφιξης έχει $n^{\Omega(\log n)}$ σημεία καμπής (κάτω φράγμα). Από την άλλη, για οποιοδήποτε γράφημα για το οποίο οι κατά τμήματα γραμμικές συναρτήσεις στις ακμές έχουν το πολύ K τμήματα συνολικά, η πολυπλοκότητα της συνάρτησης χρόνου άφιξης είναι το πολύ $n^{O(\log n)}$ (άνω φράγμα). Και τα δύο φράγματα θεωρούν ότι είτε οι ακμές σέβονται την FIFO ιδιότητα, είτε επιτρέπεται αυθαίρετη αναμονή σε κορυφές για να αντιμετωπίσουν μη-FIFO καταστάσεις στις ακμές.

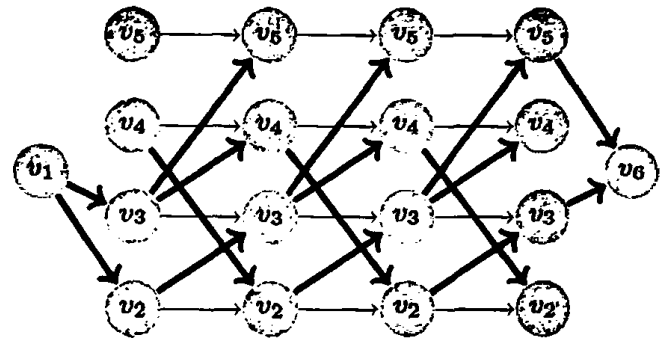
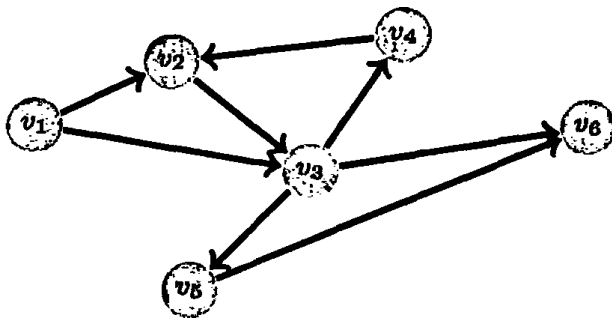
2.4.1 Μια πολυεπίπεδη αναπαράσταση του γραφήματος

Θα χρειαστούμε μια αναπαράσταση ενός άκυκλου πολυεπίπεδου κάποιου γραφήματος που αυξάνει το πλήθος των κορυφών από n σε $O(n^2)$ αλλά έχει το πλεονέκτημα ευκολότερης ανάλυσης. Αξίζει να σημειώσουμε ότι τα φράγματα που δίνουμε είναι υπερπολυωνυμικά και για το λόγο αυτό η αύξηση από n σε n^2 δεν επηρεάζει τα όρια ασυμπτωτικά.

Θεωρούμε ότι οι κορυφές του γραφήματος G αριθμούνται από 1 έως n με $s = v_1$ και $d = v_n$. Δημιουργούμε n επίπεδα αριθμημένα από 1 έως n έτσι ώστε κάθε κορυφή $v_i \notin \{s, d\}$ να αντιγράφεται σε κάθε ένα από τα $n - 2$ μεσαία επίπεδα. Το επίπεδο 1 αποτελείται μόνο από την κορυφή s και το επίπεδο n περιέχει μόνο την κορυφή d . Όλες οι ακμές του πολυεπίπεδου γραφήματος ενώνονται μεταξύ συνεχόμενων επιπέδων με κατεύθυνση από το χαμηλότερο επίπεδο προς το υψηλότερο επίπεδο. Η αφετηρία s έχει ακμές προς όλες τις (πραγματικές) γείτονες του επιπέδου 2. Αντίστοιχα όλες οι γείτονες του επιπέδου $n - 1$ με τον προορισμό d έχουν ακμές προς αυτόν. Για κάθε ακμή $(v_i, v_j) \in E(G)$ και κάθε ζεύγος γειτονικών επιπέδων τοποθετούμε ακμές των αντίστοιχων αντιγράφων του v_i και v_j . Σε κάθε τέτοια ακμή αντιστοιχούμε την ίδια συνάρτηση χρόνου άφιξης που έχει στο αρχικό γράφημα G . Επίσης εισάγουμε ακμές μεταξύ ιδίων αντιγράφων v_i που βρίσκονται σε γειτονικά επίπεδα. Οι ακμές αυτές μεταξύ των αντιγράφων έχουν συνάρτηση χρόνου άφιξης $A_{(v_i^j, v_i^{j+1})}(t) = t$ όπου v_i^j είναι το αντίγραφο της κορυφής v_i στο επίπεδο j . Δηλαδή οι τελευταίες ακμές έχουν μηδενικό χρόνο μετάβασης. Η πολυεπίπεδη αναπαράσταση με τον τρόπο αυτό εισήχθη από τους Foschini, Hershberger, και Suri [15], αν και το γράφημα του Θεωρήματος 2.1 κατασκευάζεται με παρόμοιο τρόπο αναπαράστασης [27]. Στο Σχήμα 2.5 δείχνουμε ένα παράδειγμα μετατροπής ενός γραφήματος σε μια πολυεπίπεδη αναπαράσταση.

Είναι εύκολο να παρατηρήσει κανείς ότι ένα συντομότερο μονοπάτι από το s προς το d στο αρχικό γράφημα αντιστοιχίζεται από ένα σύνολο ισοδύναμων συντομότερων μονοπατιών στην πολυεπίπεδη αναπαράσταση μόνο που κάθε τέτοιο μονοπάτι στην πολυεπίπεδη αναπαράσταση είναι άκυκλο και κάθε (s, d) -μονοπάτι έχει ακριβώς $n - 1$ ακμές. Η μετατροπή σε πολυεπίπεδη αναπαράσταση όταν έχουμε παραμετροποιημένα συντομότερα μονοπάτια είναι παρόμοια με την άνω διαδικασία με την διαφορά ότι θέτουμε μηδενικά βάρη για τις ακμές (v_i^j, v_i^{j+1}) (το γεγονός αυτό ισχύει και στην αναπαράσταση για τα χρονικά μεταβαλλόμενα





Σχήμα 2.5: Ένα γράφημα G και η αντίστοιχη πολυεπίπεδη αναπαράσταση του G . Η πολυεπίπεδη αναπαράσταση δεν έχει ποτέ κύκλους ακόμα και αν το αρχικό γράφημα G έχει κύκλους.

συντομότερα μονοπάτια ως προς τον χρόνο μετάβασης $D_{(v_i, v_{i+1})}(t)$ καθώς $A_{(v_i, v_{i+1})}(t) = t$.

Υπάρχουν δύο λόγοι για τους οποίους η πολυεπίπεδη αναπαράσταση είναι βολική ως προς τον χειρισμό της:

- Τόσο στα χρονικά μεταβαλλόμενα όσο και στα παραμετροποιημένα συντομότερα μονοπάτια η ανάλυση εστιάζει στην γραμμική συνάρτηση κόστους. Παρά την απλότητα της συνάρτησης, η γραμμική συνάρτηση είναι ταυτόχρονα τεχνητή αλλά και δύσκολα να πλαισιωθεί σωστά. Για παράδειγμα στα χρονικά μεταβαλλόμενα δίκτυα κάθε γραμμική συνάρτηση A_e με κλίση διαφορετική από 1 πρέπει να τέμνει την ευθεία $A = t$, που σημαίνει ότι $A_e - t < 0$ για $t \rightarrow \infty$ ή για $t \rightarrow -\infty$. Δηλαδή ο χρόνος μετάβασης κάποια στιγμή θα γίνει αρνητικός που έρχεται σε αντίθεση με τις αρχικές μας υποθέσεις (FIFO υπόθεση ή δυνατότητα αναμονής). Επιπλέον ένα γράφημα με αρνητικά βάρη μπορεί να περιέχει και αρνητικούς κύκλους καθιστώντας την συνάρτηση $A[s, d]$ ανούσια για κάποιες χρονικές στιγμές. Το τελευταίο γεγονός συμβαίνει και στο πρόβλημα με παραμετροποιημένα συντομότερα μονοπάτια. Η μεταφορά στην πολυεπίπεδη αναπαράσταση λύνει τα προβλήματα με τον ακόλουθο τρόπο.
- Στο πρόβλημα με παραμετροποιημένα συντομότερα μονοπάτια η πρόσθεση ενός σταθερού βάρους σε όλες τις ακμές μεταξύ δύο γειτονικών επιπέδων διατηρεί την σχετική διάταξη των (s, d) -μονοπατιών διότι κάθε τέτοιο μονοπάτι χρησιμοποιεί ακριβώς μια ακμή μεταξύ δύο γειτονικών επιπέδων. Το γεγονός αυτό θα το χρησιμοποιήσουμε κατάλληλα όταν θα επεξεργαστούμε την πολυεπίπεδη αναπαράσταση.

2.4.2 Κάτω φράγμα

Η βασική ιδέα για το κάτω φράγμα σε χρονικά μεταβαλλόμενα συντομότερα μονοπάτια είναι να χρησιμοποιήσουμε το κάτω φράγμα του Θεωρήματος 2.1 για παραμετροποιημένα συντομότερα μονοπάτια. Στην ουσία δηλαδή θέλουμε να χειριστούμε την παράμετρο γ ως την χρονική παράμετρο t . Θα ανάγουμε ένα στιγμιότυπο του παραμετροποιημένου συντομότερου μονοπατιού σε ένα στιγμιότυπο χρονικά μεταβαλλόμενου μεταβάσεων μεταφέροντας την παράμετρο γ ως την συνάρτηση χρόνου t .

Για να μπορέσει κανείς να συσχετίσει τα δύο αυτά μοντέλα θα πρέπει να αντιστοιχίσει μια συγκεκριμένη τιμή γ σε ένα εύρος χρονικών παραμέτρων χωρίς να μεταβάλει τα σημεία καμπής αρκετά και να κλιμακώσει προς τα κάτω τις καθυστερήσεις στις ακμές έτσι ώστε η παράμετρος $\gamma (= t)$ να μην αλλάζει σε μεγάλο βαθμό κατά την διάρκεια που ταξιδεύει κανείς από την αφετηρία s προς τον προορισμό d . Για τον σωστό υπολογισμό της κλιμάκωσης χρειαζόμαστε όλα τα βάρη των ακμών να είναι μη-αρνητικά αλλά όπως προείπαμε τα γραμμικά παραμετρικά βάρη είναι αρνητικά για κάποιες τιμές της παραμέτρου γ . Επομένως δοθέντος ενός γραφήματος G χειρίστης περίπτωσης από το Θεώρημα 2.1 το μετατρέπουμε πρώτα σε μια πολυεπίπεδη αναπαράσταση και στη συνέχεια θα το μετασχηματίσουμε σε ένα χρονικά μεταβαλλόμενο στιγμιότυπο.

Έστω το σύνολο των σημείων καμπής σε ένα στιγμιότυπο χειρίστης περίπτωσης παραμετροποιημένου συντομότερου μονοπατιού αντιστοιχεί στις τιμές των παραμέτρων $\gamma_1 < \gamma_2 < \dots < \gamma_N$ όπου $N = B(L[s, d])$. Υπολογίζουμε το ελάχιστο βάρος των ακμών στο εύρος $[\gamma_1, \gamma_N]$, δηλαδή $c_{\min} = \min_i \min(c_i(\gamma_1), c_i(\gamma_N))$ και τροποποιούμε το βάρος κάθε ακμής προσθέτοντας την ποσότητα $W = \max(0, -c_{\min})$. Τότε κάθε (s, d) -μονοπάτι P του αρχικού γραφήματος με το πολύ $n - 1$ ακμές και μήκος $L[P]$ αντιστοιχεί με μια σχέση ένα-προς-πολλά σε ένα σύνολο από ισοδύναμα (s, d) -μονοπάτια στην πολυεπίπεδη αναπαράσταση, όπου το κάθε ένα από αυτά τα μονοπάτια έχει ακριβώς $n - 1$ ακμές και μήκος $L[P] + (n - 1)W$. Τα σημεία καμπής της συνάρτησης $L[s, d]$ παραμένουν ίδια τόσο στο αρχικό γράφημα όσο και στην πολυεπίπεδη αναπαράσταση.

Έστω γ_j και γ_{j+1} να είναι δύο διαδοχικά σημεία καμπής της $L[s, d](\gamma)$. Συμβολίζουμε με P_j το συντομότερο (s, d) -μονοπάτι ανάμεσα στο διάστημα $[\gamma_j, \gamma_{j+1}]$. Επειδή το μήκος κάθε μονοπατιού στο G είναι μια γραμμική συνάρτηση της παραμέτρου γ , για την τιμή $\bar{\gamma}_j = (\gamma_j + \gamma_{j+1})/2$ κάθε μονοπάτι του G διαφορετικό του P_j είναι αυστηρά μεγαλύτερο από το P_j . Θα θεωρήσουμε τις ακόλουθες ποσότητες:

- $\bar{L}_j = L[s, d](\bar{\gamma}_j)$ και $\bar{L}'_j$ το μήκος του δεύτερου συντομότερου μονοπατιού στην παράμετρο $\bar{\gamma}_j$.
- $\Delta_j = \bar{L}'_j - \bar{L}_j$ και $\Delta_{\min} = \min_j \Delta_j$.

Τώρα θα υπολογίσουμε ένα μικρό ϵ -διάστημα γύρω από την παράμετρο $\bar{\gamma}_j$ τέτοιο ώστε η μεταβολή της παραμέτρου γ μέσα στο διάστημα αυτό να κρατάει την διαταραχή των βαρών σε μικρό βαθμό. Κάθε συνάρτηση κόστους $c_i(\gamma) = a_i\gamma + b_i$ έχει μια πεπερασμένη κλίση a_i . Ανάλογα με την κλίση ορίζουμε το ϵ :

- Αν $a_i \neq 0$, διαλέγουμε την παράμετρο $\epsilon_i = \Delta_{\min}/(2na_i)$ έτσι ώστε

$$|c_i(\bar{\gamma}_j) - c_i(\bar{\gamma}_j + \epsilon)| < \Delta_{\min}/(2n), \quad \text{για κάθε } \epsilon \text{ με } |\epsilon| < \epsilon_i.$$

- Αν $a_i = 0$, $|c_i(\bar{\gamma}_j) - c_i(\bar{\gamma}_j + \epsilon)| = 0 < \Delta_{\min}/(2n)$ για κάθε ϵ και επομένως μπορούμε να διαλέξουμε ϵ_i να είναι άπειρο.

Ορίζουμε ως $\epsilon_{\min} = \min_i \epsilon_i$ το οποίο είναι σταθερά που εξαρτάται μόνο από το γράφημα G και την συνάρτηση κόστους $c_i(\gamma)$.



Έστω $L_{\max}$ η μέγιστη τιμή της συνάρτησης $L[s, d](\gamma)$ στο πεπερασμένο διάστημα της παραμέτρου γ που οριοθετείται από τα σημεία καμψής, δηλαδή $\gamma \in [\gamma_1, \gamma_N]$. Ορίζουμε την σταθερά κ ως εξής:

$$\kappa = \frac{\epsilon_{\min}}{2(L_{\max} + \Delta_{\min})}.$$

Στη συνέχεια μετατρέπουμε το πρόβλημα του παραμετροποιημένου συντομότερου μονοπατιού σε ένα χρονικά μεταβαλλόμενο θέτοντας τον (χρονικά μεταβαλλόμενο) χρόνο μετάβασης $D_{e_i} = \kappa c_i(t)$, δηλαδή κανονικοποιούμε τα κόστη κατά κ και ισοδυναμούμε την χρονική μεταβολή t με την παράμετρο γ . Θυμίζουμε ότι $A_e(t) = D_e(t) + t$.

Τώρα μπορεί να δείξει κανείς ότι ο χρόνος μετάβασης κάθε ακμής στο γράφημα G μεταβάλλεται κατά το πολύ $\kappa \Delta_{\min} / (2n)$ στο χρονικό διάστημα $t \in [\bar{\gamma}_j, \bar{\gamma}_j + \epsilon_{\min})$, που αποτελεί το χρονικά διάστημα για να επισκεφτεί το μονοπάτι P_j ξεκινώντας την χρονική στιγμή $t = \bar{\gamma}_j$. Δηλαδή το μονοπάτι P_j είναι το συντομότερο (s, d) -μονοπάτι με χρόνο εκκίνησης $t = \bar{\gamma}_j$. Επειδή αυτό ισχύει για κάθε j μεταξύ του 1 και $N - 1$ υπάρχουν τουλάχιστον $N - 2$ συνδυασμοί αλλαγών στο συντομότερο (s, d) -μονοπάτι στο χρονικά μεταβαλλόμενο γράφημα μεταξύ χρονικών στιγμών $t = \gamma_1$ και $t = \gamma_N$. Επομένως το ακόλουθο αποτέλεσμα προκύπτει λαμβάνοντας υπόψιν το Θεώρημα 2.1.

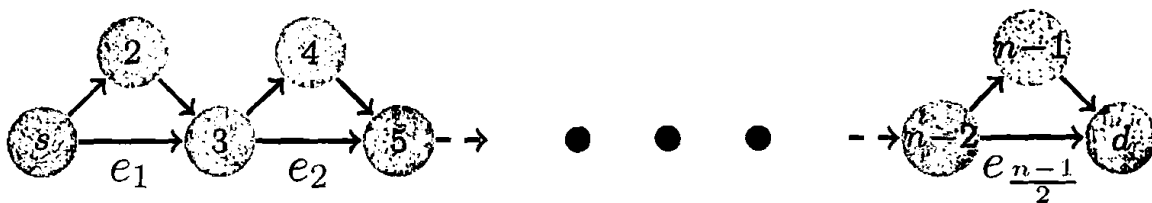
Θεώρημα 2.2 ([15]). Υπάρχει γράφημα με γραμμικές συναρτήσεις χρόνου άφιξης στις ακμές A_e και δύο κορυφές s και d τέτοιο ώστε το πλήθος των σημείων καμψής του $A[s, d]$ είναι $B(A[s, d]) = n^{\Omega(\log n)}$.

2.4.3 Άνω φράγμα

Σε αυτή την ενότητα θα δείξουμε ότι το πλήθος των σημείων καμψής για οποιαδήποτε συνάρτηση χρόνου άφιξης σε ένα γράφημα είναι $K \times n^{O(\log n)}$, όπου K είναι το συνολικό πλήθος (και τμημάτων) των γραμμικών συναρτήσεων που περιγράφουν τους χρόνους άφιξης σε όλες τις ακμές του γραφήματος.

Μια πρώτη εκτίμηση είναι ότι τα σημεία καμψής μπορεί να είναι το πολύ σε πλήθος όσο και οι διαφορετικές κλίσεις των συναρτήσεων σε όλα τα δυνατά (s, d) -μονοπάτια. Δυστυχώς η εκτίμηση αυτή δεν δίνει χρήσιμα άνω φράγματα, καθώς το πλήθος των διαφορετικών κλίσεων σε όλα τα δυνατά (s, d) -μονοπάτια μπορεί να είναι της τάξης του πλήθους των διαφορετικών (s, d) -μονοπατιών. Για παράδειγμα αν θεωρήσει κανείς το γράφημα του Σχήματος 2.6 στο οποίο κάθε μια από της $\Theta(n)$ ακμές e_i μπορεί να επιλεγεί από μόνη της να ανήκει σε κάθε (s, d) -μονοπάτι. Αν η συνάρτηση χρόνου άφιξης A_{e_i} είναι η κάθε μια γραμμική με την κλίση της ίση με την τιμή a_i , όπου $a_i = i$ -στός μικρότερος πρώτος αριθμός -1 , και για κάθε άλλη ακμή e έχουμε $A_e(t) = t$ (δηλαδή, μηδενική καθυστέρηση) τότε το πλήθος των διαφορετικών κλίσεων σε όλες τις συναρτήσεις $A[P]$ με $P \in \mathcal{P}_{s,d}$ είναι $2^{\Theta(n)}$. Αυτό εξηγείται από το γεγονός ότι οι κλίσεις των συναρτήσεων $A[P]$ είναι γινόμενα διαφορετικών υποσυνόλων πρώτων αριθμών και συνεπώς διαφορετικές μεταξύ τους.

Για τον υπολογισμό της πολυπλοκότητας $A[s, d]$ θα θεωρήσουμε πρώτα την περίπτωση στην οποία κάθε συνάρτηση χρόνου άφιξης A_e είναι γραμμική. Όπως αναφέραμε προηγουμένως στην περίπτωση αυτή θα πρέπει να μετατρέψουμε το αρχικό γράφημα G σε μια



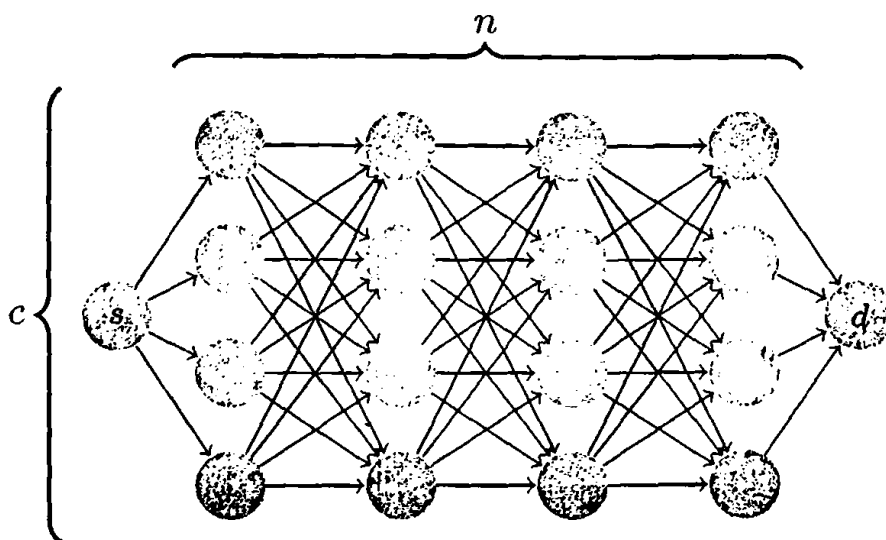
Σχήμα 2.6: Κάθε μια από τις e_i ακμές ανήκει σε διαφορετικό (s, d) -μονοπάτι.

πολυεπίπεδη αναπαράσταση για να αποφύγουμε αρνητικούς κύκλους. Συμβολίζουμε με $A_{lin}[s, d]$ μια συνάρτηση συντομότερης άφιξης μεταξύ όλων των (s, d) -μονοπατιών θεωρώντας ότι κάθε A_e είναι μια γραμμική συνάρτηση (και όχι κατά τμήματα γραμμική συνάρτηση). Τότε $B(A_{lin}[s, d])$ είναι το μέγιστο πλήθος των σημείων καμπής στην αντίστοιχη συνάρτηση συντομότερης άφιξης για γραμμικές συναρτήσεις χρόνου άφιξης σε κάθε ακμή της πολυεπίπεδης αναπαράστασης και για το πλήθος αυτό υπάρχει το ακόλουθο άνω φράγμα.

Λήμμα 2.2 ([15]). Για το πλήθος των σημείων καμπής στην συνάρτηση συντομότερης άφιξης στα (s, d) -μονοπάτια ισχύει:

$$B(A[s, d]) \leq K \times B(A_{lin}[s, d]).$$

Τώρα για να φράξουμε το $B(A_{lin}[s, d])$ θα χρησιμοποιήσουμε ένα αποτέλεσμα από τον Gusfield [19]. Πιο συγκεκριμένα θα χρησιμοποιήσουμε μια αναδρομική πολυεπίπεδη αναπαράσταση $G_{n,c}$ που είναι ένα γράφημα με $nc + 2$ κορυφές που αποτελείται από δύο διακεκριμένες κορυφές s και d και n στήλες που περιέχουν c κορυφές η κάθε στήλη. Οι ακμές του $G_{n,c}$ ενώνουν την κορυφή s με τις κορυφές της πρώτης στήλης, ή ενώνουν κορυφές σε διαδοχικές στήλες i και $i + 1$, ή ενώνουν τις κορυφές της τελευταίας στήλης n με την κορυφή d . Στο Σχήμα 2.7 δείχνουμε ένα τέτοιο παράδειγμα.



Σχήμα 2.7: Το γράφημα $G_{n,c}$ με την πολυεπίπεδη αναπαράστασή του.

Λήμμα 2.3 ([15]). Για το γράφημα $G_{n,c}$ με γραμμικές συναρτήσεις χρόνου στις ακμές, ισχύει:

$$B(A_{lin}[s, d]) \leq \frac{(2n+1)^{1+\log c}}{2}.$$

Στη συνέχεια κατασκευάζουμε την πολυεπίπεδη αναπαράσταση $G_{n-1,n-1}$ από το αρχικό γράφημα G όπως περιγράφηκε προηγουμένως. Από την ισοδυναμία των (s, d) -μονοπατιών μεταξύ της πολυεπίπεδης αναπαράστασης και του αρχικού γραφήματος, συμπεραίνουμε με βάση το Λήμμα 2.3 ότι το $B(A_{lin}[s, d])$ φράσσεται άνω από $\frac{(2n+1)^{1+\log n}}{2}$ και στα δύο γραφήματα. Επομένως για οποιοδήποτε γράφημα G με γραμμικές συναρτήσεις χρόνου άφιξης A_e σε κάθε ακμή $e \in E(G)$ έχουμε $B(A_{lin}[s, d]) = n^{O(\log n)}$. Έτσι εφαρμόζοντας το Λήμμα 2.2 καταλήγουμε στο ακόλουθο άνω φράγμα.

Θεώρημα 2.3 ([15]). Για οποιοδήποτε γράφημα G όπου οι συναρτήσεις χρόνου άφιξης A_e σε κάθε ακμή $e \in E(G)$ περιγράφονται από κατά τμήματα γραμμικές συναρτήσεις με συνολικά K τμήματα για όλες τις ακμές ισχύει $B(A[s, d]) = K \times n^{O(\log n)}$ για οποιοδήποτε ζεύγος κορυφών s, d . Αν το K είναι πολυωνυμικό ως προς το πλήθος κορυφών n , τότε $B(A[s, d]) = n^{O(\log n)}$.



ΚΕΦΑΛΑΙΟ 3

ΑΛΓΟΡΙΘΜΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΓΙΑ ΥΠΟΛΟΓΙΣΜΟΥΣ ΣΥΝΑΡΤΗΣΕΩΝ ΣΥΝΤΟΜΟΤΕΡΗΣ ΑΦΙΞΗΣ

3.1 Αλγόριθμος Χρονικά Εξαρτώμενος από την Έξοδο (output-sensitive)

3.2 Προσέγγιση Συναρτήσεων με εκ των Προτέρων Γνώση

3.3 Προσεγγιστικός Αλγόριθμος

3.4 Προσεγγιστικός Αλγόριθμος για Άγνωστες Συναρτήσεις

Στο κεφάλαιο αυτό θα παρουσιάσουμε ορισμένα αλγοριθμικά αποτελέσματα για το πρόβλημα ΣΥΝΑΡΤΗΣΕΙΣ ΣΥΝΤΟΜΟΤΕΡΗΣ ΑΦΙΞΗΣ, ΓΙΑ ΑΝΑΧΩΡΗΣΗ ΑΠΟ ΔΕΔΟΜΕΝΗ ΑΦΕΤΗΡΙΑ (ΣΣΑ) όπου κάθε συνάρτηση χρόνου άφιξης $A_e(t)$ περιγράφεται από κατά τμήματα γραμμικές συναρτήσεις. Ειδικότερα θα δείξουμε πρώτα έναν αλγόριθμο που υπολογίζει (και κατασκευάζει) την συνάρτηση συντομότερης άφιξης $A[s, d]$ έτσι ώστε η χρονική πολυπλοκότητα να εξαρτάται όχι από το χειρόστης-περίπτωσης μέγεθος, αλλά από το πραγματικό μέγεθος της συνάρτησης συντομότερης άφιξης. Δηλαδή ο χρόνος εκτέλεσης του αλγορίθμου είναι εξαρτώμενος από το μέγεθος του εξόδου που υπολογίζει (output-sensitive algorithm). Πριν από τους προσεγγιστικούς αλγορίθμους θα περιγράψουμε έναν αλγόριθμο που προσεγγίζει με ελάχιστο πλήθος σημείων καμπής μια κατά τμήματα γραμμική συνάρτηση έχοντας, ωστόσο, γνώση της συνάρτησης $D[s, d]$. Στη συνέχεια θα δείξουμε έναν πολυωνυμικού-χρόνου εκτέλεσης προσεγγιστικό αλγόριθμο που κατασκευάζει μια προσέγγιση (μικρού μεγέθους) για την συνάρτηση του χρόνου μετάβασης.



3.1 Αλγόριθμος Χρονικά Εξαρτώμενος από την Έξοδο (output-sensitive)

Τα κάτω και άνω φράγματα που παρουσιάσαμε στην Ενότητα 2.4, δείχνουν ότι η πολυπλοκότητα της συνάρτησης $A[s, d]$ μπορεί να φτάσει να γίνει $n^{\Theta(\log n)}$. Ωστόσο, στην πράξη η πολυπλοκότητα αυτή μπορεί να είναι πολύ μικρότερη και επομένως ένας αλγόριθμος που υπολογίζει την $A[s, d]$ σε χρόνο ανάλογο ως προς το πλήθος των αλλαγών των συντομότερων μονοπατιών είναι επιθυμητός. Θα παρουσιάσουμε έναν τέτοιο αλγόριθμο χρονικά εξαρτώμενο από το πλήθος των σημείων καμπής που εμφανίζονται σε κορυφές και ακμές του γραφήματος G για ένα δεδομένο σύνολο κατά τμήματα γραμμικών συναρτήσεων που περιγράφουν τις συναρτήσεις χρόνου άφιξης. Ο αλγόριθμος θα υπολογίζει την συνάρτηση $A[s, v]$ για κάθε κορυφή $v \in V(G)$ συμπεριλαμβανομένου και του προορισμού d όπως περιγράφεται στην εργασία [15].

Γενικά η ιδέα βασίζεται στο να τρέξουμε πρώτα τον αλγόριθμο του Dijkstra για μια δεδομένη χρονική στιγμή t (Αλγόριθμος 2) και να κατασκευάσουμε ένα σύνολο από πιστοποιητικά (certificates, κατηγορήματα γραμμικά εξαρτώμενα από τον χρόνο t) που μπορούν να εγγυηθούν την ορθότητα του δέντρου συντομότερων μονοπατιών που κατασκευάζει ο αλγόριθμος του Dijkstra. Στη συνέχεια μεταβάλλουμε την χρονική στιγμή αναχώρησης t ενώ παράλληλα διατηρούμε το δέντρο συντομότερων μονοπατιών και το σύνολο πιστοποιητικών για την εξασφάλιση της ορθότητας. Η τεχνική αυτή παράγει όλα τα διαφορετικά δέντρα συντομότερων μονοπατιών που υπάρχουν για κάθε τιμή του χρόνου t , δημιουργώντας κάθε ένα δέντρο από μια μεταβολή ενός παρόμοιου γειτονικού δέντρου.

Τα πιστοποιητικά είναι δύο ειδών, *πιστοποιητικά ακμών* και *πιστοποιητικά κορυφών*.

- Τα πιστοποιητικά ακμών προσδιορίζονται από το επόμενο πρωταρχικό σημείο καμπής σε κάποια ακμή. Η αποτυχία ενός πιστοποιητικού ακμής αντιστοιχεί σε ένα πρωταρχικό σημείο καμπής για κάποια ακμή.
- Τα πιστοποιητικά κορυφών εξαρτώνται από την εφαρμογή του τελεστή $\min$ στις συναρτήσεις χρόνου άφιξης μέσω των εισερχόμενων ακμών σε κάποια κορυφή. Αποτυχία ενός πιστοποιητικού κορυφής αντιστοιχεί σε μια καμπή ελαχιστοποίησης.

Κάθε πιστοποιητικό έχει μια χρονική στιγμή αποτυχίας t_{fail} που είναι ο συντομότερος χρόνος αναχώρησης για τον οποίο το πιστοποιητικό αποτυγχάνει μετά την τρέχουσα στιγμή t .

Τα πιστοποιητικά αποθηκεύονται σε μια ουρά προτεραιότητας και αυτό με το συντομότερο χρόνο αποτυχίας τοποθετείται στην κεφαλή. Όταν ένα πιστοποιητικό αποτυγχάνει, τότε (i) μια τουλάχιστον κορυφή ή ακμή του γραφήματος G έχει σημείο καμπής, (ii) το συντομότερο μονοπάτι από την αφετηρία s προς κάποια ή κάποιες άλλες κορυφές πιθανόν να αλλάξει, και (iii) κάποια πιστοποιητικά πρέπει να ενημερωθούν κατάλληλα.

3.1.1 Προεπεξεργασία γραφήματος

Όταν πρέπει να ενημερώσουμε τα πιστοποιητικά κορυφών θα πρέπει να περιγράφουμε με αποτελεσματικό τρόπο τις αντίστοιχες αλλαγές. Για να περιγράψουμε καλύτερα τα πιστο-



ποιητικά κορυφών θα προεπεξεργαστούμε το γράφημα έτσι ώστε κάθε κορυφή να έχει έσω-βαθμό το πολύ 2 (δηλαδή, στο τροποποιημένο γράφημα θέλουμε να ισχύει $\text{indgree}(v) \leq 2$ για κάθε κορυφή v).

Για κάθε κορυφή $v \in V(G)$ σχηματίζουμε ένα δυαδικό δέντρο όπου τα φύλλα του είναι οι εισερχόμενες ακμές (u, v) . Το δέντρο αυτό αντιπροσωπεύει μια αναμέτρηση (tournament) ανάμεσα στις εισερχόμενες ακμές για να καθορίσει εκείνη την ακμή που ανήκει στο συντομότερο μονοπάτι για να φτάσει κανείς στην v από την αφετηρία s σε οποιαδήποτε χρονική στιγμή. Για τον λόγο αυτό (και μόνο για λόγους απλότητας), τροποποιούμε το γράφημα G επεκτείνοντας κάθε κορυφή v με έσω-βαθμό $k = \text{indgree}(v)$ σε ένα δυαδικό δέντρο με $k - 1$ κόμβους, ύψους $\lceil \log_2 k \rceil$, και κάθε εσωτερική ακμή του δέντρου έχει μηδενική καθυστέρηση. Αυτή η μετατροπή επιφέρει το γράφημα G' με $\Theta(m)$ κορυφές και κάθε κορυφή έχει έσω-βαθμό το πολύ δύο.

Στο εξής μελετάμε και λαμβάνουμε υπόψιν το τροποποιημένο γράφημά μας G' όπου κάθε κορυφή έχει έσω-βαθμό το πολύ δύο.

3.1.2 Περιγραφή του αλγορίθμου

Εφαρμόζουμε τον αλγόριθμο του Dijkstra (Αλγόριθμος 2) στο γράφημα G' ξεκινώντας από την αφετηρία s κάποια χρονική στιγμή t_0 . Ο αλγόριθμος υπολογίζει μια γραμμική συνάρτηση σε κάθε κορυφή v που αναπαριστά $A[s, v]$ για το χρονικό διάστημα που περιέχει την στιγμή t_0 . Έστω ότι η κορυφή v έχει δύο προκατόχους u_1 και u_2 τέτοιοι ώστε το δέντρο συντομότερου μονοπατιού για την χρονική στιγμή αναχώρησης t_0 περιέχει την ακμή $e_1 = (u_1, v)$ ενώ δεν περιέχει την ακμή $e_2 = (u_2, v)$. Από την ορθότητα του αλγορίθμου του Dijkstra έχουμε ότι $A[s, v|e_1](t_0) < A[s, v|e_2](t_0)$, δηλαδή το συντομότερο μονοπάτι από την αφετηρία s την χρονική στιγμή t_0 και φτάνοντας στην κορυφή v μέσω της ακμής e_1 φτάνει νωρίτερα από το συντομότερο μονοπάτι την ίδια χρονική στιγμή αναχώρησης μέσω της ακμής e_2 . Για τα πιστοποιητικά θα έχουμε:

- Υπολογίζουμε το πιστοποιητικό κορυφής για την κορυφή v με την εύρεση της χρονικής στιγμής $t_{\min} > t_0$ (αν υπάρχει τέτοια) για την οποία η γραμμική συνάρτηση $A[s, v|e_1]$ γίνεται ίση με την $A[s, v|e_2]$. Ο χρόνος αποτυχίας του πιστοποιητικού θα είναι $t_{\min}$ ή ∞ αν οι δύο γραμμικές συναρτήσεις δεν τέμνονται μετά την χρονική στιγμή t_0 .
- Ένα πιστοποιητικό ακμής υπολογίζεται για κάθε ακμή $e = (u, v)$ χρησιμοποιώντας την αντίστροφη της γραμμικής συνάρτησης $A[s, u]$ για την προβολή του επόμενου πρωταρχικού σημείου καμπής της A_e (αν υπάρχει τέτοιο) στον χώρο του χρόνου της αφετηρίας s . Το αποτέλεσμα από αυτή την διαδικασία θα είναι μια χρονική στιγμή $t_e > t_0$ που αποτελεί τον χρόνο αποτυχίας για το συγκεκριμένο πιστοποιητικό.

Τα $m + n$ συνολικά πιστοποιητικά των κορυφών και ακμών του G' τοποθετούνται σε μια ουρά προτεραιότητας ταξινομημένα ως προς τον επόμενο χρόνο αποτυχίας κάθε πιστοποιητικού. Όταν το πρώτο πιστοποιητικό αποτύχει κάποια χρονική στιγμή t_{fail} , η γραμμική συνάρτηση $A[s, v]$ αλλάζει για κάποιες κορυφές v .



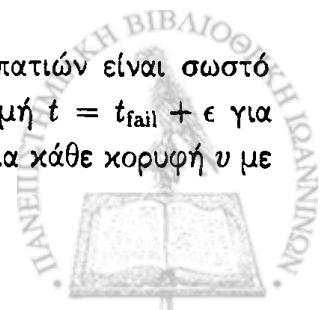
- Αν το πιστοποιητικό κορυφής αποτύχει στην κορυφή v (δηλαδή, το συντομότερο μονοπάτι μέσω της ακμής (u_1, v) γίνει πιο αργό από το συντομότερο μονοπάτι μέσω της ακμής (u_2, v)) τότε το δέντρο συντομότερων μονοπατιών αλλάζει στην κορυφή v . Την ίδια στιγμή η συναρτησιακή μορφή της $A[s, v]$ αλλάζει όπως επίσης και κάθε συνάρτησης $A[s, w]$ στο τρέχον δέντρο συντομότερων μονοπατιών όπου w είναι ένας απόγονος του v στο γράφημα G' . Επίσης χρήζουν ενημέρωσης και τα πιστοποιητικά ακμών που εξέρχονται από κορυφές $w \in T_v(t_0)$ καθώς επίσης και τα πιστοποιητικά κορυφών που αντιστοιχούν στις κεφαλές των εξερχόμενων ακμών.
- Με παρόμοιο τρόπο, αν ένα πιστοποιητικό ακμής για την ακμή (u, v) στο δέντρο συντομότερων μονοπατιών αλλάζει τότε η συναρτησιακή μορφή κάθε συνάρτησης $A[s, w]$ αλλάζει για την κορυφή v και κάθε κορυφή w που είναι απόγονος της v στο $T_v(t_0)$. Επίσης χρειάζονται ενημέρωση τα πιστοποιητικά ακμών εκτός δέντρου που εξέρχονται από κορυφές του $T_v(t_0)$ καθώς και τα πιστοποιητικά των κεφαλών τους. Αν αλλάξει το πιστοποιητικό ακμής για μια ακμή (u, v) που δεν βρίσκεται στο δέντρο συντομότερων μονοπατιών τότε το πιστοποιητικό κορυφής για την κορυφή v χρειάζεται να ανανεώσει τον χρόνο αποτυχίας του αλλά καμία άλλη αλλαγή δεν χρειάζεται να εφαρμοστεί.

Σε κάθε περίπτωση αν η συναρτησιακή μορφή $A[s, v|e]$ αλλάζει για κάποια ακμή $e = (u, v)$ τότε πρέπει το πιστοποιητικό κορυφής για την κορυφή v να ενημερώσει τον χρόνο αποτυχίας του.

Όταν μια γραμμική συνάρτηση $A[s, v]$ αλλάζει για κάποια κορυφή v αλλά δεν αλλάζει για τον προκάτοχο της v στο δέντρο συντομότερων μονοπατιών τότε πρέπει να ενημερώσουμε τα πιστοποιητικά για όλους τους απόγονους της v στο δέντρο συντομότερων μονοπατιών. Στο σημείο αυτό εφαρμόζουμε τα βήματα που περιγράφονται από την ακόλουθη διαδικασία:

1. Διασχίζουμε το υποδέντρο ριζωμένο στην v ενημερώνοντας κατάλληλα την συναρτησιακή μορφή της $A[s, w]$ για κάθε απόγονο w .
2. Επαναυπολογίζουμε τους χρόνους αποτυχίας
 - των πιστοποιητικών ακμών για ακμές που είναι προσβάσιμες από την κορυφή v και
 - των πιστοποιητικών κορυφών για κορυφές που είναι προσβάσιμες από την κορυφή v και για κεφαλές ακμών που δεν ανήκουν στο δέντρο και είναι προσβάσιμες από την κορυφή v .
3. Ενημερώνουμε την ουρά προτεραιότητας που αποθηκεύει τα πιστοποιητικά για να λάβουμε υπόψη νέες χρονικές στιγμές αποτυχίας.

Στο τέλος της διαδικασίας αυτής (i) το δέντρο συντομότερων μονοπατιών είναι σωστό για μονοπάτια που αναχωρούν από την αφετηρία s την χρονική στιγμή $t = t_{\text{fail}} + \epsilon$ για απειροελάχιστο ϵ , (ii) η συναρτησιακή μορφή της $A[s, v]$ είναι σωστή για κάθε κορυφή v με



χρόνο αναχώρησης $t = t_{\text{fail}} + \epsilon$, και (iii) η ουρά προτεραιότητας αναγνωρίζει την επόμενη χρονική στιγμή μετά την t_{fail} στην οποία μια αλλαγή σε μια ή περισσότερες συναρτήσεις άφιξης χρόνου θα συμβεί.

Αξιίζει να σημειώσουμε ότι για το δέντρο συντομότερων μονοπατιών από την αφετηρία s έχουμε τις εξής ιδιότητες:

- Σε κάθε αποτυχία ενός πιστοποιητικού κορυφής το δέντρο αλλάζει μόνο κατά μια ακμή.
- Σε κάθε αποτυχία ενός πιστοποιητικού ακμής το δέντρο παραμένει αμετάβλητο.

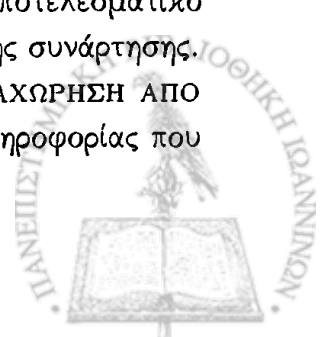
3.1.3 Ανάλυση χρονικής πολυπλοκότητας

Το συνολικό πλήθος των πιστοποιητικών είναι $O(m)$ διότι κάθε πιστοποιητικό ανήκει σε μια το πολύ κορυφή και ακμή. Ο χρόνος που χρειαζόμαστε για να αντιμετωπίσουμε σε μια χρονική αποτυχία ενός πιστοποιητικού c είναι $O(|T_c| \log n)$, όπου T_c είναι το σύνολο ακμών $e = (u, v)$ του γραφήματος G' για τις οποίες οι συναρτήσεις χρόνου άφιξης $A[s, v|e]$ μεταβάλλονται εξαιτίας της αποτυχίας του πιστοποιητικού c και ο λογαριθμικός όρος εμφανίζεται εξαιτίας των λειτουργιών της ουράς προτεραιότητας. Το σύνολο ακμών T_c αποτελείται από υποδέντρα συντομότερων μονοπατιών στο γράφημα G' ριζωμένα στην κορυφή ή στην ακμή για την οποία το πιστοποιητικό c εμφανίζεται, με επιπρόσθετα τις ακμές που δεν ανήκουν στο δέντρο κατευθυνόμενες προς τα έξω από το υποδέντρο. Στην χειρότερη περίπτωση $|T_c|$ μπορεί να είναι $\Theta(m)$ αλλά σε περιπτώσεις που είναι μικρότερο ο χρονικά εξαρτώμενος αλγόριθμος από την έξοδο θα πρέπει να συμπεριλάβει το γεγονός αυτό. Εκφραζόμενο ως προς το αρχικό γράφημα G , $|T_c|$ ισοδυναμεί με το πλήθος των εξερχόμενων ακμών που είναι προσβάσιμα από το πιστοποιητικό της κορυφής ή ακμής που βρίσκονται στο υποδέντρο επί έναν όρο το πολύ $O(\log n)$ που αντιστοιχεί στο ύψος των δέντρων στα οποία αυτές οι ακμές συμμετέχουν στο γράφημα G' .

Η χρονική πολυπλοκότητα του αλγορίθμου στη χειρότερη περίπτωση είναι $O(\log^2 n)$ επί το συνολικό πλήθος των σημείων καμπής των $A[s, v|e]$ για κάθε ακμή $e = (u, v) \in E(G)$. Από τους δύο λογαριθμικούς όρους ο ένας εμφανίζεται εξαιτίας της ουράς προτεραιότητας και ο άλλος λογαριθμικός όρος υπάρχει εξαιτίας της δομής δεδομένων που διαλέξαμε για το $|T_c|$.

3.2 Προσέγγιση Συναρτήσεων με εκ των Προτέρων Γνώση

Ο αλγόριθμος που θα δούμε στην Ενότητα 3.3 μπορεί να προσεγγίσει οποιαδήποτε άγνωστη συνάρτηση που περιγράφεται από κατά τμήματα γραμμικές συναρτήσεις (μέσω του προβλήματος ΣΣΑ) χωρίς γνώση της συνάρτησης χρόνου μετάβασης $D[s, d]$ με αποτελεσματικό τρόπο, δηλαδή περιγράφοντας με όσον το δυνατό λιγότερα σημεία καμπής της συνάρτησης. Φυσικά στο πρόβλημα ΣΥΝΑΡΤΗΣΕΙΣ ΣΥΝΤΟΜΟΤΕΡΗΣ ΑΦΙΞΗΣ, ΓΙΑ ΑΝΑΧΩΡΗΣΗ ΑΠΟ ΔΕΔΟΜΕΝΗ ΑΦΕΤΗΡΙΑ (ΣΣΑ) δεν έχουμε εκ των προτέρων γνώση της πληροφορίας που



περιγράφεται από την συνάρτηση $D[s, d]$. Ωστόσο αν είχαμε γνώση της συνάρτησης $D[s, d]$ είναι ενδιαφέρον να προσεγγίσει μια κατά τμήματα γραμμική συνάρτηση με τα ελάχιστα δυνατά σημεία καμπής. Με αυτό ακριβώς το πρόβλημα θα ασχοληθούμε στην ενότητα αυτή περιγράφοντας έναν αποτελεσματικό αλγόριθμο προσέγγισης μιας κατά τμήματα γραμμικής συνάρτησης. Σημειώνουμε ότι ο αλγόριθμος αυτός δεν μπορεί να εφαρμοστεί στο πρόβλημα ΣΣΑ καθώς χρειάζεται γνώση εκ των προτέρων της συνάρτησης $D[s, d]$ και παραθέτουμε την ενότητα αυτή για λόγους πληρότητας.

Έστω $p_1 = (x_1, y_1), \dots, p_n = (x_n, y_n)$ σημεία στο επίπεδο $\mathbb{R}^2$ τέτοια ώστε $x_1 < \dots < x_n$. Έστω $f : [x_1, x_n] \rightarrow \mathbb{R}$ να είναι μια συνάρτηση της οποίας το γράφημα $y = f(x)$ στο επίπεδο είναι πολύγωνη ευθεία που συνδέει τα σημεία του χώρου $p_1, \dots, p_n$. Τότε η συνάρτηση f είναι κατά τμήματα γραμμική συνάρτηση και x -μονότονη.

Τέτοιου είδους συνάρτηση f μπορεί να προσεγγιστεί από μια άλλη κατά τμήματα γραμμική συνάρτηση g με μικρότερο αριθμό τμημάτων K_g . Το πρόβλημα το οποίο θέλουμε να εξετάσουμε είναι το εξής:

ΠΡΟΣΕΓΓΙΣΗ ΚΑΤΑ ΤΜΗΜΑΤΑ ΓΡΑΜΜΙΚΗΣ ΣΥΝΑΡΤΗΣΗΣ

Είσοδος: Μια συνάρτηση f που περιγράφεται από n σημεία

$p_1 = (x_1, y_1), \dots, p_n = (x_n, y_n)$ και ένα σφάλμα προσέγγισης $\epsilon > 0$

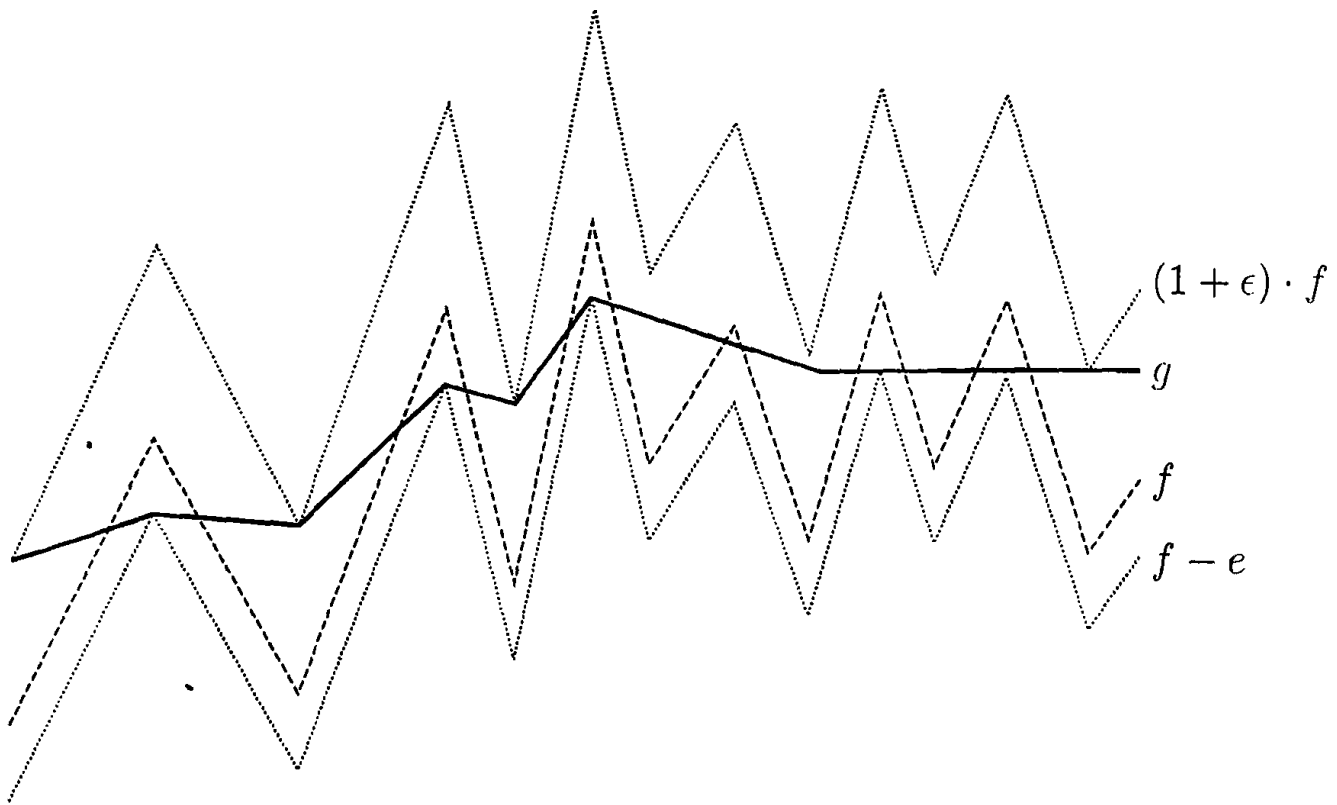
Ερώτημα: Εύρεση της συνάρτησης g τέτοια ώστε $(1 - \epsilon)f \leq g \leq (1 + \epsilon)f$ και K_g να είναι το ελάχιστο δυνατό.

Υπάρχουν διάφοροι τρόποι για να ορίσει κανείς τα όρια σφάλματος μεταξύ της δοθέν συνάρτησης και της προσεγγιστικής. Ένας τέτοιος τρόπος στοχεύει στον υπολογισμό ενός άνω ή κάτω φράγματος της δοθέν συνάρτησης. Αλλά επίσης προσεγγίσεις που βρίσκονται μέσα στο φάσμα μεταξύ του κάτω και άνω φράγματος μπορεί κανείς να αναζητήσει. Δηλαδή είναι ενδιαφέρον να ορίσει κανείς μόνο ένα από τα δύο φράγματα στο πρόβλημα. Μια άλλη παραλλαγή του προβλήματος είναι τα σχετικά σφάλματα να μετατραπούν σε απόλυτα σφάλματα $f - \epsilon$ και $f + \epsilon$.

Στο Σχήμα 3.1 δείχνουμε μια κατά τμήματα γραμμική συνάρτηση f και μια προσέγγισή της g που βρίσκεται μέσα στο φάσμα μεταξύ των δύο ορίων. Το άνω φράγμα ορίζεται ως ένα σχετικό άνω σφάλμα ϵ της f ενώ το κάτω φράγμα ορίζεται από ένα απόλυτο σφάλμα e της f . Η συνάρτηση f έχει 14 δειγματοληπτικά σημεία ενώ η συνάρτηση προσέγγισης g αποτελείται από $K_g = 8$ τμήματα που είναι και το ελάχιστο δυνατό.

Το πρόβλημα στο οποίο θα επικεντρωθούμε βασίζεται στον υπολογισμό άνω φραγμάτων που ικανοποιούν σχετικά σφάλματα. Για το πρόβλημα αυτό υπάρχουν διάφοροι αλγόριθμοι που βασίζονται σε ευρετικά κριτήρια (αν δεν ικανοποιείται η μονοτονία ως προς τις x -συντεταγμένες) [10, 36] ή σε γεωμετρικές ιδιότητες των δειγματοληπτικών σημείων [23, 34]. Θα αναφερθούμε στον αλγόριθμο των Imai και Iri που τρέχει σε χρόνο $O(n)$ και είναι βέλτιστος ως προς την πολυπλοκότητα εξόδου του προβλήματος [23]. Αξίζει να σημειώσουμε ότι πειραματική μελέτη του αλγορίθμου έχει υλοποιηθεί με αποτελεσματικό τρόπο [29].

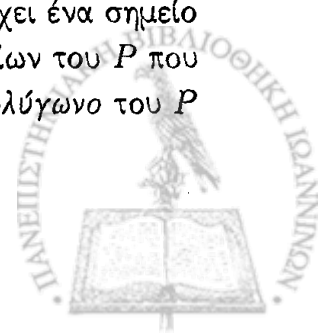
Η βασική ιδέα του αλγορίθμου [23] είναι ο υπολογισμός ενός ελάχιστου μονοπατιού μέσα σε ένα “τούνελ” μεταξύ των δύο κατά τμήματα γραμμικών συναρτήσεων. Στην περίπτωση



Σχήμα 3.1: Άνω και κάτω φράγματα μιας κατά τμήματα γραμμικής συνάρτησης f που ορίζονται από σχετικό σφάλμα ϵ και από απόλυτο σφάλμα e , αντίστοιχα.

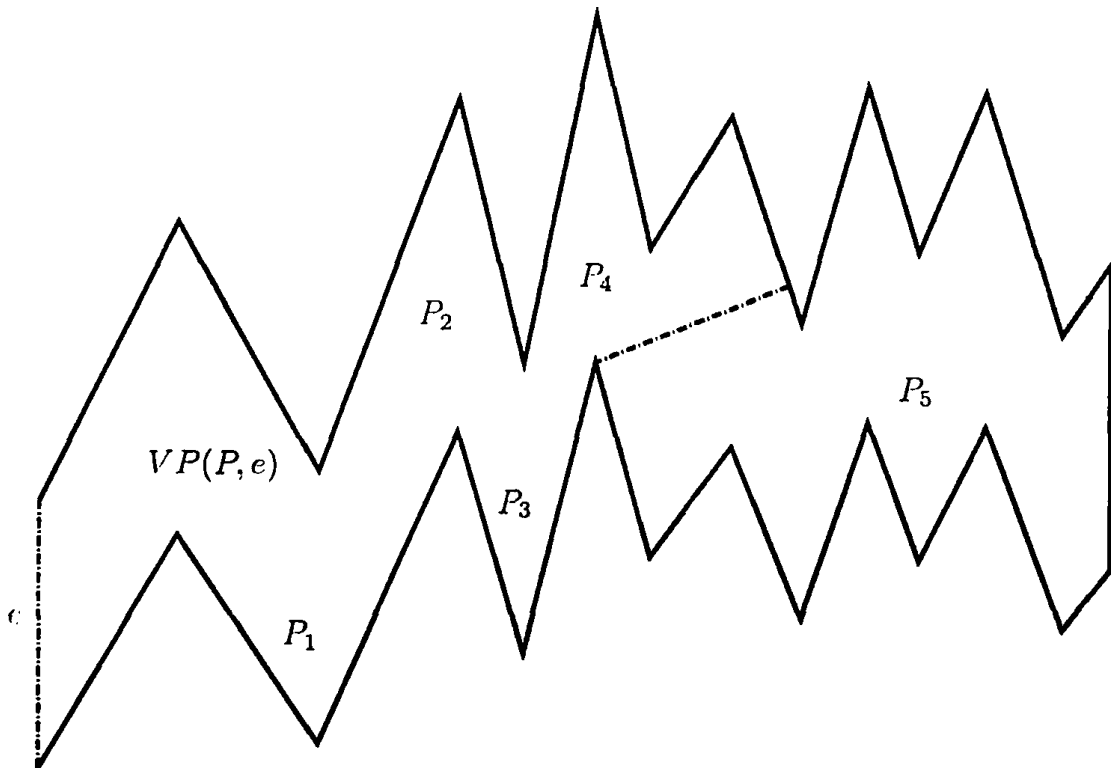
μας το τούνελ χτίζεται μεταξύ της συνάρτησης f και της συνάρτησης του άνω ορίου $(1+\epsilon)f$. Φανταζόμαστε μια πηγή φωτός (ένας φακός) να έχει τοποθετηθεί στην είσοδο του τούνελ. Αυτή η πηγή φωτίζει ένα μέρος του τούνελ και χωρίζει το τούνελ σε ένα ορατό κομμάτι και σε διάφορα μη-ορατά μέρη. Κάθε τέτοιο κομμάτι (ορατό ή μη-ορατό) περιβάλλεται από κάποια όρια μεταξύ των δύο συναρτήσεων και από κάποια κοινά σημεία. Η τομή του ορίου του ορατού κομματιού και του μη-ορατού κομματιού που περιέχει την έξοδο του τούνελ καλείται παράθυρο. Στο παράθυρο θα ακουμπήσει το πρώτο σημείο της προσεγγιστική συνάρτησης που ψάχνουμε. Μετακινούμε τον φακό στο πρώτο παράθυρο που υπολογίσαμε και στη συνέχεια υπολογίζουμε τα υπόλοιπα κομμάτια με παρόμοιο τρόπο έως ότου το παράθυρο τοποθετηθεί στην έξοδο του τούνελ. Την βασική ιδέα την δείχνουμε παραστατικά στο Σχήμα 3.2.

Για να τυποποιήσουμε τον αλγόριθμο θα χρειαστούμε ορισμένες έννοιες σχετικές με την ορατότητα. Έστω P το πολύγωνο (δηλαδή το τούνελ) που οριοθετείται μεταξύ των δύο συναρτήσεων. Δύο σημεία p, q του P θα λέμε ότι είναι ορατά αν το ευθύγραμμο τμήμα που τα ενώνει βρίσκεται μέσα στο πολύγωνο P . Έστω e_p ένα ευθύγραμμο τμήμα του P που βρίσκεται στα όρια του P . Ένα σημείο p του πολύγωνου P θα λέμε ότι είναι ορατό από το e_p αν υπάρχει ένα σημείο στο ευθύγραμμο τμήμα e_p που είναι ορατό με το p . Παρόμοια, ένα ευθύγραμμο τμήμα e'_p είναι ορατό από ένα άλλο e_p αν υπάρχει ένα σημείο στο ευθύγραμμο τμήμα e'_p που είναι ορατό από το e_p . Το σύνολο των σημείων του P που είναι ορατά από τουλάχιστον ένα σημείο του e_p καλείται ασθενώς ορατό πολύγωνο του P .



ως προς το e_P και συμβολίζεται ως $VP(P, e_P)$. Αξιζει να σημειώσουμε ότι η ορατότητα από το e_P διαχωρίζει το πολύγωνο P σε $VP(P, e_P)$ και κάποια άλλα μη-ορατά πολύγωνα από το e_P .

Στο Σχήμα 3.2 τα μη-ορατά πολύγωνα από το e είναι τα P_1, P_2, P_3, P_4, P_5 . Το σκιασμένο κομμάτι αποτελεί το ασθενώς ορατό πολύγωνο $VP(P, e)$ από την είσοδο e του πολυγώνου P .



Σχήμα 3.2: Η βασική ιδέα του Αλγορίθμου 6 με ένα παράδειγμα εκτέλεσης.

Στον Αλγόριθμο 6 δείχνουμε τα βήματα του αλγορίθμου που περιγράψαμε με τις σχετικές έννοιες. Το πρώτο πολύγωνο P_1 αρχικοποιείται στο περίβλημα των συναρτήσεων και το αντίστοιχο παράθυρο τοποθετείται στην είσοδο του τούνελ. Όσο η έξοδος του τούνελ $[p_n^+, p_n^-]$ δεν είναι ορατή από το τρέχον παράθυρο υπολογίζουμε νέο παράθυρο και, αντίστοιχα, νέο πολύγωνο. Κάθε φορά που υπολογίζουμε ένα παράθυρο και ένα πολύγωνο διαλέγουμε ένα σημείο της προσεγγιστικής συνάρτησης που ψάχνουμε το οποίο βρίσκεται στην τομή της ευθείας μεταξύ του νέου παραθύρου e_{i+1} και του παραθύρου e_i . Η διαδικασία αυτή σταματά κατά την έξοδο από το τούνελ οπότε και υπολογίζουμε τα τελευταία δύο σημεία της προσεγγιστικής συνάρτησης που ψάχνουμε.

Οι Imai και Iri έδειξαν ότι ο αλγόριθμος αυτός σωστά υπολογίζει την ελάχιστη προσεγγιστική συνάρτηση και τρέχει σε $O(n)$ χρόνο [23]. Αν κανείς υπολογίσει τα πολύγωνα απευθείας τότε θα χρειαστεί χρόνο $O(K_g n \log n)$ που εξαρτάται από την έξοδο. Για να αποφύγουν τον άμεσο υπολογισμό των πολυγώνων χρησιμοποίησαν γραμμικό αλγόριθμο για τον υπολογισμό κυρτού πολυγώνου [35]. Επίσης βασίστηκαν σε γεωμετρικές ιδιότητες για τον υπολογισμό των παραθύρων σχετικές με κυρτά πολύγωνα που σχηματίζουν οι

Αρχικοποίηση:

for $i = 1$ **to** n **do**

$p_i^- \leftarrow p_i$ /* κάτω φράγμα για το τούνελ */

$p_i^+ \leftarrow (x_i, (1 + \epsilon)y_i)$ /* άνω φράγμα για το τούνελ */

$P_1 \leftarrow$ πολύγωνο $p_1^-, p_2^-, \dots, p_n^-, p_n^+, p_{n-1}^+, \dots, p_1^+$

$e_1 \leftarrow [p_1^+, p_1^-]$

$i \leftarrow 1$

Βασικός βρόγχος:

while $[p_n^+, p_n^-]$ δεν είναι ορατό από e_i στο πολύγωνο P_i **do**

$e_{i+1} \leftarrow$ παράθυρο από e_i προς το $[p_n^+, p_n^-]$ στο πολύγωνο P_i

$P_{i+1} \leftarrow$ πολύγωνο μη-ορατό από e_i που περιέχει το $[p_n^+, p_n^-]$ στο πολύγωνο P_i

$q_i \leftarrow$ σημείο τομής της ευθείας που περιέχει το e_{i+1} και e_i

$i \leftarrow i + 1$

$K_g \leftarrow i + 1$

$(q_{K_g-1}, q_{K_g}) \leftarrow$ δύο σημεία στην e_i και στην $[p_n^+, p_n^-]$ που είναι ορατά μεταξύ τους στο πολύγωνο P_{K_g-1}

$g \leftarrow q_1, q_2, \dots, q_{K_g}$

Αλγόριθμος 6: Τούνελ με φακό

συναρτήσεις f και το άνω φράγμα της f .

3.3 Προσεγγιστικός Αλγόριθμος

Στο επόμενο κομμάτι του κεφαλαίου θα ασχοληθούμε με το πρόβλημα ΣΣΑ. Παρόλο που η πολυπλοκότητα της συνάρτησης συντομότερης άφιξης $A[s, d]$ στη χειρότερη περίπτωση είναι $n^{\Theta(\log n)}$, στην πράξη μπορεί κανείς να αρκεστεί με μια μικρότερης πολυπλοκότητας προσέγγιση της συνάρτησης συντομότερης άφιξης. Για να ποσοτικοποιήσουμε το σφάλμα προσέγγισης, θα χρησιμοποιήσουμε την συνάρτηση χρόνου μετάβασης $D(t) = A[s, d](t) - t$. Θα λέμε ότι μια συνάρτηση $D'(t)$ είναι $(1 + \epsilon)$ -προσέγγιση της $D(t)$, αν $|D(t) - D'(t)| \leq \epsilon \cdot D(t)$, $\forall t \geq 0$.

Αν κανείς κοιτάξει το διάστημα αναχωρήσεων δύο διαδοχικών πρωταρχικών εικόνων τότε μπορεί να παρατηρήσει ότι η συνάρτηση $A[s, d](t)$ σχηματίζει μια κοίλη αλυσίδα (concave chain) [15]. Επομένως και η συνάρτηση $D(t)$ είναι κοίλη μεταξύ δύο γειτονικών πρωταρχικών εικόνων. Επειδή το πλήθος των πρωταρχικών σημείων καμπής είναι $O(K)$, ανάγουμε το πρόβλημα της προσέγγισης της συνάρτησης $D(t)$ σε $O(K)$ προσεγγιστικά προβλήματα για κοίλες αλυσίδες. Ενώ δεν έχουμε απευθείας πρόσβαση στις κοίλες αλυσίδες, μπορούμε να τις ανιχνεύσουμε με αποτελεσματικό τρόπο χρησιμοποιώντας ευθεία και προς-τα-πίσω εκτελέσεις του Dijkstra αλγορίθμου. Αυτό γίνεται για να βρούμε τις πρωταρχικές εικόνες χωρίς να γνωρίζουμε τις συναρτήσεις χρόνου άφιξης έτσι ώστε να τις



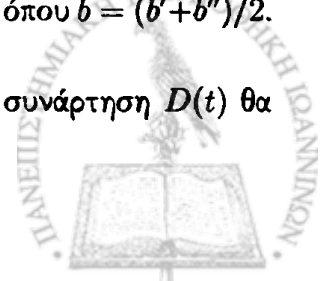
αντιστρέφουμε.

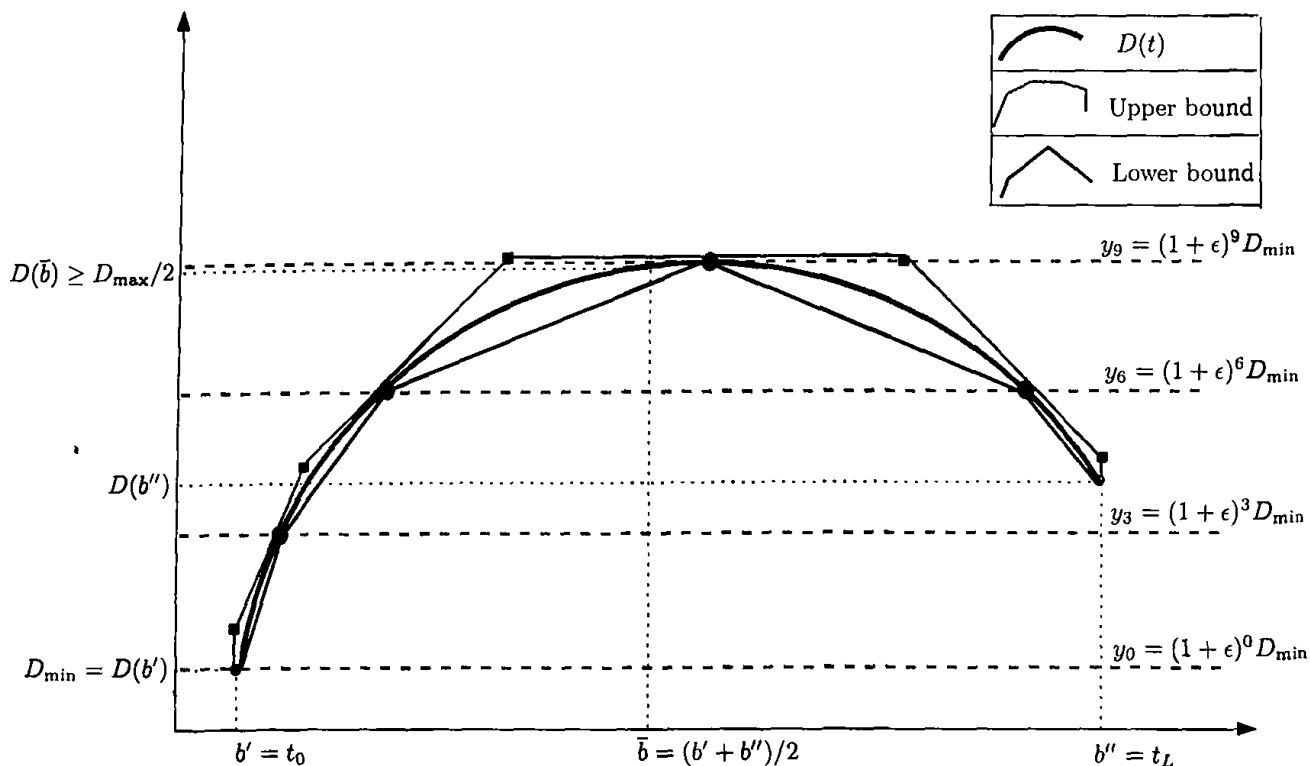
Μια ευθεία αναζήτηση με βάση τον αλγόριθμο του Dijkstra αποτελείται από μια εκτέλεση του TD-Dijkstra αλγορίθμου για μια δεδομένη χρονική στιγμή αναχώρησης t και καθορίζει την αντίστοιχη συνάρτηση συντομότερης άφιξης $A[s, d](t)$. Επίσης μια προς-τα-πίσω αναζήτηση εκτελεί τον TD-Dijkstra αλγόριθμο στο ανεστραμμένο γράφημα $\overleftarrow{G}$ και αντιστρέφει τις συναρτήσεις χρόνου άφιξης $A_e(t)$ για κάθε ακμή e , έτσι ώστε δοθείσης κάποιας χρονικής στιγμής άφιξης a στον προορισμό d , καθορίζει την χρονική στιγμή t (ύστατης αναχώρησης) για την οποία $A[s, d](t) = a$. Οι πιθανές ασυνέχειες της συνάρτησης $A_e(t)$ αντιστοιχούν σε οριζόντιες ακμές με συνάρτηση $A_{\neq}(t)$ και αντίστροφα. Για παράδειγμα αν μια ακμή e έχει μη-συνεχή συνάρτηση $A_e(t)$, δηλαδή για δύο τιμές τ^-, τ^+ ισχύει $A_e(\tau^-) = x$ και $A_e(\tau^+) = y$ με $x < y$, τότε στη συνάρτηση $A_{\neq}(t)$ κάθε τιμή στο εύρος $[x, y]$ αντιστοιχεί η τιμή τ . Αυτό εξηγείται από το γεγονός ότι ταξιδεύοντας από την αφετηρία s κάποιος πρέπει να καταλήξει στην κεφαλή της ακμής e την χρονική στιγμή $t = \tau$ ή νωρίτερα έτσι ώστε να φτάσει το άλλο άκρο της ακμής e το πολύ την χρονική στιγμή που βρίσκεται στο εύρος $[x, y]$.

Το πρώτο βήμα της προσέγγισης είναι να κομματιάσουμε την συνάρτηση $A[s, d]$ και επομένως και την $D(t)$, σε όλες τις πρωταρχικές εικόνες. Αντί να προσπαθήσουμε να βρούμε ακριβώς εκείνα τα πρωταρχικά σημεία καμπής με αντίστοιχες εικόνες στην συνάρτηση $A[s, d]$, θα κομματιάσουμε την συνάρτηση $A[s, d]$ στις εικόνες όλων των πρωταρχικών σημείων καμπής, ακόμα και σε αυτά τα σημεία που δεν συνεισφέρουν στην συνάρτηση $A[s, d]$. Για κάθε ακμή $e = (u, v)$ με σημείο καμπής την χρονική στιγμή b ως προς το σημείο αναφοράς της u , προβάλλουμε τον χρόνο b πίσω στο βοηθητικό σημείο αναφοράς της αφετηρίας s ως εξής: εκτελούμε μια προς-τα-πίσω αναζήτηση με τον αλγόριθμο TD-Dijkstra στο γράφημα $\overleftarrow{G}$ και αντιστρέφοντας τις συναρτήσεις χρόνου άφιξης για κάθε ακμή. Αυτό μας επιστρέφει μια χρονική στιγμή b' ως προς το σημείο αναφοράς της s έτσι ώστε αν αναχωρήσει κανείς την στιγμή b' τότε θα χτυπήσει το σημείο καμπής της e την χρονική στιγμή b . Δηλαδή, η χρονική αυτή στιγμή είναι η πρωταρχική εικόνα στην συνάρτηση $A[s, d]$ που αποτελεί και την εικόνα του χρόνου b . Η ένωση όλων αυτών των πρωταρχικών εικόνων συμβολίζεται ως PI και αποτελεί ένα υπερσύνολο (καθώς μια χρονική στιγμή b' μπορεί να μην ανήκει απαραίτητα στην συνάρτηση $A[s, d]$) των πρωταρχικών εικόνων της συνάρτησης $A[s, d]$.

Στη συνέχεια για κάθε ζεύγος διαδοχικών σημείων καμπής $b', b'' \in PI$ εκτελούμε μια ευθεία αναζήτηση με τον αλγόριθμο TD-Dijkstra για να βρούμε τις αντίστοιχες τιμές της συνάρτησης $A[s, d]$ και επομένως τις τιμές των συναρτήσεων μετάβασης $D(b')$ και $D(b'')$, αντίστοιχα. Η συνάρτηση $D(t)$ είναι κοίλη μεταξύ των σημείων b' και b'' . Δηλαδή για κάθε τριάδα t_1, t_2, t_3 με $b' < t_1 < t_2 < t_3 < b''$ το σημείο με συντεταγμένες $(t_2, D(t_2))$ βρίσκεται ακριβώς ή πάνω από το ευθύγραμμο τμήμα μεταξύ των σημείων με συντεταγμένες $(t_1, D(t_1))$ και $(t_3, D(t_3))$. Η ελάχιστη τιμή της συνάρτησης $D(t)$ στο διάστημα $[b', b'']$ είναι $D_{\min} = \min(D(b'), D(b''))$ και η μέγιστη τιμή $D_{\max}$ είναι το πολύ $2D(\bar{b})$ όπου $\bar{b} = (b' + b'')/2$. Στο Σχήμα 3.3 δείχνουμε σχηματικά τα σημεία αυτά.

Αν γνωρίζαμε τη συνάρτηση $D[s, d]$ τότε μια προσέγγιση για την συνάρτηση $D(t)$ θα





Σχήμα 3.3: Τα άνω και κάτω φράγματα προσέγγισης για την $D(t)$ στο τμήμα $[b', b'']$.

μπορούσε να γίνει μέσω οριζόντιων τμημάτων (slices): βρισκουμε την τομή της $D(t)$ με τις ευθείες

$$y = (1 + \epsilon)^k D_{\min}$$

για κάθε $k \geq 0$ τέτοιο ώστε $(1 + \epsilon)^k D_{\min} \leq D_{\max}$. Έστω ότι τα σημεία τομής που έχουν προκύψει από τις οριζόντιες τομές έχουν συντεταγμένες $(t_i, D(t_i))$. Επειδή η συνάρτηση $D(t)$ είναι κοίλη στο διάστημα $[b', b'']$, κάθε οριζόντια τομή επιφέρει το πολύ δυο δειγματοληπτικά σημεία. Επομένως ο συνολικός αριθμός των δειγματοληπτικών σημείων είναι το πολύ $2 \log(D_{\max}/D_{\min})/\log(1 + \epsilon) \approx \frac{2}{\epsilon} \log(D_{\max}/D_{\min})$.

- Κάτω Φράγμα Προσέγγισης: Η γραμμική παρεμβολή μεταξύ των διαδοχικών σημείων $(t_i, D(t_i))$ δίνει μια προσέγγιση που δεν μπορεί να είναι μεγαλύτερη από την $D(t)$.
- Άνω Φράγμα Προσέγγισης: Το μικρότερο περίβλημα των οριζόντιων τμημάτων σε όλα τα δειγματοληπτικά σημεία είναι μια κοίλη καμπύλη που δεν μπορεί να είναι μικρότερη από την $D(t)$.

Το σφάλμα στη χειρότερη περίπτωση και στις δύο προσεγγίσεις είναι το πολύ ϵ επί την πραγματική τιμή της συνάρτησης $D(t)$, δίνοντας έτσι μια $(1 + \epsilon)$ -προσέγγιση της $D(t)$ όπως επιθυμούμε. Τα δύο φράγματα δείχνονται στο Σχήμα 3.3. Το αποτέλεσμα αυτό περιγράφεται στο ακόλουθο λήμμα.

Λήμμα 3.1 ([15]). Αν οι συναρτήσεις χρόνου άφιξης έχουν συνολικά K γραμμικά τμήματα και η μέγιστη τιμή της ποσότητας $D_{\max}/D_{\min}$ σε όλα τα κοίλα διαστήματα της $D[s, d]$

είναι R , τότε υπάρχει μια δομή δεδομένων μεγέθους $O(K \frac{1}{\epsilon} \log R)$ που αναπαριστά την συνάρτηση $D[s, d]$ με σχετικό σφάλμα ϵ .

Αυτό που έμεινε για να περιγράψουμε πλήρως τον προσεγγιστικό αλγόριθμο είναι ο τρόπος με τον οποίο παίρνουμε δειγματοληπτικά σημεία για να προσεγγίσουμε την συνάρτηση $D(t)$. Σημειώνουμε ότι δεν μπορούμε να εφαρμόσουμε απευθείας τις οριζόντιες διατομές της $D(t)$ διότι έχουμε πρόσβαση μόνο στη συνάρτηση $A[s, d](t)$ και όχι στην $D(t)$. Ωστόσο μπορούμε να προσεγγίσουμε καλά χωρίς να εντοπίσουμε επακριβώς την χρονική στιγμή t_i για την οποία $D(t_i) = (1 + \epsilon)^i D_{\min}$ για κάθε $i \leq 2 \log(D_{\max}/D_{\min})/\log(1 + \epsilon)$. Αρκεί να προσεγγίσουμε δειγματοληπτικά σημεία σε μια λεπτότερη διαμέριση του εύρους τιμών. Δηλαδή, φανταζόμαστε ότι τεμαχίζουμε την $D(t)$ σε περιοχές με τις οριζόντιες ευθείες $y = (1 + \epsilon)^{k/2} D_{\min}$ (χρησιμοποιώντας το διπλάσιο πλήθος τιμών για το k) και στη συνέχεια βρίσκουμε ένα δειγματοληπτικό σημείο σε κάθε περιοχή. Οι τιμές της συνάρτησης $D(t)$ σε διαδοχικά δειγματοληπτικά σημεία διαφέρουν το πολύ κατά έναν παράγοντα του $(1 + \epsilon)$ και επομένως αρκούν για να ικανοποιήσουν τα κριτήρια του Λήμματος 3.1.

Επομένως επικεντρωνόμαστε στο πρόβλημα της εύρεσης προσεγγιστικά δειγματοληπτικά σημεία για μια κοίλη αλυσίδα που οριοθετείται μεταξύ δύο διαδοχικών πρωταρχικών σημείων καμπής b' και b'' . Έστω ότι το οριζόντιο εύρος της i -στης αλυσίδας συμβολίζεται με $I_i = b'' - b'$. Το κάθετο εύρος οριοθετείται μεταξύ του $D_{\min}$ και $D_{\max}$. Το συνολικό οριζόντιο εύρος όλων των K αλυσίδων συμβολίζεται με $I = \sum_{1 \leq i \leq K} I_i$. Σημειώνουμε ότι η κλίση της $D(t)$ είναι μονότονα φθίνουσα μέσα στο διάστημα $[b'', b']$ και από την FIFO ιδιότητα η κλίση της $D(t)$ στο σημείο $t = b''$ είναι τουλάχιστον -1 .

Χωρίζουμε την διαδικασία που αναλύει τα σημεία καμπής του διαστήματος $[b', b'']$ σε δύο φάσεις: στην πρώτη φάση όσο η κλίση της συνάρτησης $D(t)$ είναι τουλάχιστον 1 επεξεργαζόμαστε σημεία που προκύπτουν από ευθείες και προς-τα-πίσω αναζητήσεις, ενώ στη δεύτερη φάση όσο η κλίση της συνάρτησης $D(t)$ είναι το πολύ 1 εφαρμόζουμε διχοτόμηση σε σημεία που προκύπτουν από ευθείες αναζητήσεις.

Πιο συγκεκριμένα, θα χρησιμοποιήσουμε δύο αναζητήσεις (ευθεία και προς-τα-πίσω) με βάση τον αλγόριθμο TD-Dijkstra όπως περιγράψαμε προηγουμένως. Σημειώνουμε ότι οι προς-τα-πίσω αναζητήσεις έχουν ασθενή συσχέτιση με την συνάρτηση $D(t)$ καθώς δεν γνωρίζουμε την χρονική στιγμή t εξαρχής. Στην πρώτη φάση αναζητούμε δειγματοληπτικά σημεία πάνω στην συνάρτηση $D(t)$ χρησιμοποιώντας προς-τα-πίσω αναζητήσεις πάνω σε ένα πρόθεμα του διαστήματος $[b', b'']$ και στη δεύτερη φάση χρησιμοποιούμε ευθείες αναζητήσεις στο υπόλοιπο κομμάτι του διαστήματος.

Ξεκινώντας στην πρώτη φάση την στιγμή $t = b'$ και όσο η κλίση της συνάρτησης $D(t)$ είναι τουλάχιστον 1 εφαρμόζουμε προς-τα-πίσω εκτελέσεις του TD-Dijkstra με επιθυμητή τιμή άφιξης στον προορισμό

$$a = t + D(t) \cdot \sqrt{1 + \epsilon}$$

και στη συνέχεια θέτουμε την χρονική στιγμή t ίση με το αποτέλεσμα της προς-τα-πίσω αναζήτησης t' . Επειδή $t' \geq t$ θα έχουμε

$$D(t') = A(t') - t' \leq A(t') - t = D(t) \cdot \sqrt{1 + \epsilon}.$$



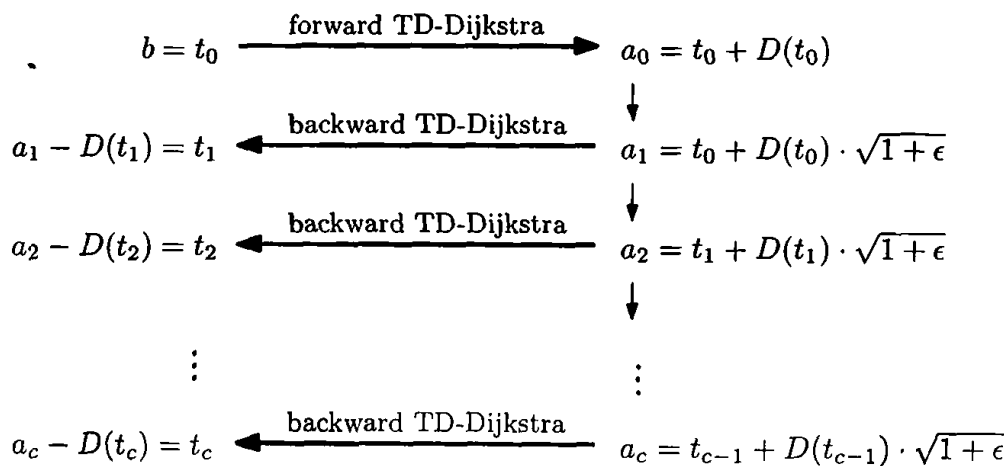
Δηλαδή το σημείο $(t', D(t'))$ βρίσκεται στο τρέχον τμήμα ή στο επόμενο. Αν ωστόσο η κλίση της $D(t')$ είναι τουλάχιστον 1 τότε η συνάρτηση $A[s, d]$ έχει κλίση τουλάχιστον 2 μεταξύ του t και t' , που σημαίνει ότι

$$a = t + D(t) \cdot \sqrt{1 + \epsilon} \geq t + D(t) + 2(t' - t).$$

Επομένως $t' \leq \frac{1}{2}D(t)(\sqrt{1 + \epsilon} - 1) + t$ που σημαίνει ότι

$$D(t') = t + D(t)\sqrt{1 + \epsilon} - t' \geq \frac{1}{2}D(t)(\sqrt{1 + \epsilon} + 1).$$

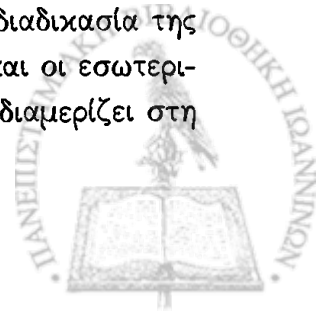
Αλλά τότε η ποσότητα $(\sqrt{1 + \epsilon} + 1)/2$ είναι ένα σταθερό ποσοστό της ποσότητας $\sqrt{1 + \epsilon}$ και επομένως αυτή η διαδικασία εκτελεί ένα σταθερό αριθμό από προς-τα-πίσω αναζητήσεις σε κάθε τμήμα. Στο Σχήμα 3.4 δείχνουμε σχηματικά την επαναληπτική διαδικασία της πρώτης φάσης.



Σχήμα 3.4: Η επαναληπτική διαδικασία υπολογισμού των σημείων της πρώτης φάσης όσο η κλίση της συνάρτησης $D(t)$ είναι τουλάχιστον 1.

Στη δεύτερη φάση της διαδικασίας, όταν οι προς-τα-πίσω αναζητήσεις παράγουν ένα διάστημα $[t, b'']$ με την κλίση της συνάρτησης $D(t)$ να είναι το πολύ 1 τότε χρησιμοποιούμε διχοτόμηση (bisection, δηλαδή 'κάθετες ευθείες') με ευθείες αναζητήσεις για να βρούμε τα υπόλοιπα δειγματοληπτικά σημεία. Σημειώνουμε ότι, όπως θα δείξουμε στη συνέχεια, αν οι κλίσεις της συνάρτησης $D(t)$ στα άκρα ενός διαστήματος μήκους ℓ διαφέρουν κατά Δ τότε το μέγιστο κάθετο σφάλμα του διαστήματος φράσσεται από $\ell\Delta$. Η επακριβής απόδειξη του προηγούμενου ισχυρισμού δίνεται στο Λήμμα 3.3. Έστω ότι το διάστημα μήκους ℓ έχει διαφορά κλίσης Δ και η ελάχιστη τιμή για την συνάρτηση $D(t)$ είναι $\bar{D}$ (καθοριζόμενη σε ένα από τα δύο άκρα του διαστήματος). Αν $\ell\Delta > \bar{D}(1 + \epsilon)$, τότε διχοτομούμε το διάστημα στο μεσαίο t με μια ευθεία αναζήτηση και αναδρομικά διαμερίζουμε τα δυο υποδιαστήματα. Αυτή η διαδικασία δίνει προσεγγιστικό σφάλμα το πολύ $1 + \epsilon$ μεταξύ διαδοχικών σημείων.

Στη συνέχεια φράσσουμε το πλήθος των αναζητήσεων που προκύπτουν από την διχοτόμηση. Θεωρούμε το δυαδικό δέντρο αναζήτησης που προκύπτει από την διαδικασία της διχοτόμησης. Τα φύλλα του δέντρου αναπαριστούν τα τελικά διαστήματα και οι εσωτερικοί κόμβοι του δέντρου αναπαριστούν τις διχοτομήσεις. Κάθε διχοτόμηση διαμερίζει στη



μέση το τρέχον διάστημα. Μετά από k διχοτομήσεις το τρέχον υποδιάστημα έχει μήκος $I/2^k$ και επομένως το μέγιστο σφάλμα για αυτό το υποδιάστημα είναι το πολύ $2I/2^k$ διότι $\Delta \leq 2$. Αυτό σημαίνει ότι το ύψος του δέντρου είναι το πολύ $\log(2I/(\epsilon D_{\min}))$. Οι εσωτερικοί κόμβοι του δέντρου στο μέγιστο βάθος του δέντρου αντιστοιχούν σε διαστήματα με $\ell\Delta > \bar{D}(1 + \epsilon)$. Επομένως σε οποιαδήποτε $(1 + \epsilon)$ -προσέγγιση του αρχικού διαστήματος τουλάχιστον ένα σημείο προέρχεται από ένα τέτοιο διάστημα. Από το Λήμμα 3.1 γνωρίζουμε ότι υπάρχουν το πολύ $O(\frac{1}{\epsilon} \log(D_{\max}/D_{\min}))$ προσεγγιστικά σημεία στη χειρότερη περίπτωση. Το επιπλέον κόστος για την εύρεση κάθε προσεγγιστικού σημείου χρησιμοποιώντας την διαδικασία διχοτόμησης είναι το πλήθος το διχοτομήσεων από την ρίζα σε ένα φύλλο στο δέντρο διχοτόμησης. Η ποσότητα αυτή φράσσεται από το ύψος του δέντρου. Οι αντίστοιχες ανισότητες στο άθροισμα λογαρίθμων επεκτείνουν το αποτέλεσμα για τα K συνολικά τμήματα των γραμμικών συναρτήσεων διότι

$$\sum_{1 \leq i \leq K} \log\left(\frac{I_i}{\epsilon D_{\min}}\right) \leq K \log\left(\frac{I}{K\epsilon D_{\min}}\right).$$

Επομένως το συνολικό πλήθος αναζητήσεων με τον TD-Dijkstra αλγόριθμο δίνεται από το ακόλουθο αποτέλεσμα.

Θεώρημα 3.1 ([15]). *Το συνολικό πλήθος των αναζητήσεων συντομότερων μονοπατιών που χρειάζονται για να υπολογιστεί η προσέγγιση του Λήμματος 3.1 είναι*

$$O\left(K \frac{1}{\epsilon} \log(D_{\max}/D_{\min}) \log\left(\frac{I}{K\epsilon D_{\min}}\right)\right).$$

3.4 Προσεγγιστικός Αλγόριθμος για Άγνωστες Συναρτήσεις

Στην παρούσα ενότητα δίνουμε μια πιο ακριβής ανάλυση του προσεγγιστικού αλγορίθμου η οποία βασίζεται στο λεπτομερή υπολογισμό του μέγιστου σφάλματος που μπορεί να προκύψει σε κάθε βήμα του αλγορίθμου. Η γνώση αυτού του μέγιστου σφάλματος μας επιτρέπει και τον σχεδιασμό ενός ακριβέστερου προσεγγιστικού αλγορίθμου, αλλά και την παράκαμψη των περιττών δειγμάτων κατά την πρώτη φάση ώστε να ελαχιστοποιηθεί (ασυμπτωτικά) η πολυπλοκότητα (δηλαδή, το πλήθος των σημείων καμπής) της παραγόμενης συνάρτησης.

Πρώτα δείχνουμε ότι η μέγιστη τιμή της $D(t)$ δεν μπορεί να υπερβεί την διπλάσια τιμή της D στο μέσο του διαστήματος $T = [\tau_0, \tau_N]$.

Λήμμα 3.2.

$$D_{\max} \equiv \max_{t \in [\tau_0, \tau_N]} D(t) < 2 \cdot D\left(\frac{\tau_0 + \tau_N}{2}\right).$$

Απόδειξη. Γνωρίζουμε ότι η συνάρτηση $D(t)$ παίρνει θετικές τιμές και είναι κοίλα. Παρατηρούμε ότι για οποιοδήποτε σημείο $t \in [\tau_0, \tau_N]$, το συμμετρικό σημείο $\bar{t} = (\tau_0 + \tau_N) - t$ ως προς το $\frac{\tau_0 + \tau_N}{2}$ βρίσκεται μέσα στο διάστημα $[\tau_0, \tau_N]$. Επομένως για κάθε $t, \bar{t} \in [\tau_0, \tau_N]$ θα έχουμε:

$$\frac{D(t)}{2} < \frac{D(t) + D(\bar{t})}{2} \leq D\left(\frac{t + \bar{t}}{2}\right) = D\left(\frac{\tau_0 + \tau_N}{2}\right) \Rightarrow D(t) < 2D\left(\frac{\tau_0 + \tau_N}{2}\right).$$

■

Αντίστοιχα η ελάχιστη τιμή της συνάρτησης $D(t)$ στο διάστημα $[\tau_0, \tau_N]$ είναι $D_{\min} = \min(D(\tau_0), D(\tau_N))$. Στη συνέχεια θα θεωρήσουμε ότι για οποιαδήποτε χρονική στιγμή $t \in T = [\tau_0, \tau_N]$ έχουμε πρόσβαση στην πληροφορία που περιγράφεται από την τριάδα $(a^-(t), a^+(t), D(t))$ όπου οι τιμές $a^-(t), a^+(t)$ περιγράφουν την δεξιά και αριστερή παράγωγο (κλίση) της $D(t)$.

Όπως ήδη αναφέραμε στόχος μας είναι να κατασκευάσουμε την προσεγγιστική συνάρτηση $D'(t)$ της $D(t)$ σε πολυωνυμικό χρόνο με όσο το δυνατό λιγότερα σημεία καμπής. Ο αλγόριθμος θα κατασκευάσει μια κατά τμήματα γραμμική προσέγγιση $D'(t)$ της $D(t)$ της οποίας το πλήθος των σημείων καμπής (που θα το αναφέρουμε και ως πολυπλοκότητα της συνάρτησης) θα είναι ανεξάρτητο από το αντίστοιχο της $D(t)$ και θα εξαρτάται από την ποιότητα της ζητούμενης προσέγγισης και το μέγεθος του ελάχιστου τετραγώνου που περικλείει την συνάρτηση (δηλαδή, το τετράγωνο αυτό περικλείεται από τις ευθείες $D_{\max} - D_{\min}$ και $\tau_N - \tau_0$).

Η περιγραφή του αλγορίθμου βασίζεται στον αλγόριθμο προσέγγισης που δόθηκε από τους Foschini et. al. [15]. Αν και στην εργασία [15] ο αλγόριθμος προσεγγίζει μια κατά τμήματα γραμμική συνάρτηση σε ένα χρονικά μεταβαλλόμενο δίκτυο, ο αλγόριθμος που θα περιγράψουμε προσεγγίζει οποιαδήποτε συνεχή κοίλα συνάρτηση με ελάχιστη κλίση ≥ -1 για την οποία υποθέτουμε ότι μας δίνονται ορισμένες τιμές της άγνωστης συνάρτησης μαζί με τις αντίστοιχες κλίσεις της συνάρτησης. Για τον λόγο αυτό θα θεωρήσουμε ότι

- η $f(t)$ είναι η άγνωστη συνεχή κοίλα συνάρτηση (σε αντιστοιχία με την κατά τμήματα γραμμική $A[s, d](t)$),
- $\tilde{f}(t)$ η προσεγγιστική συνάρτηση που ψάχνουμε (δηλαδή θέλουμε να ισχύει $|f(t) - \tilde{f}(t)| \leq \epsilon t$),
- $g(t) = t + f(t)$ (σε αντιστοιχία με την συνάρτηση $D(t)$ ως προς την $A[s, d](t)$),
- $g^{-1} : g(t) \mapsto t$ η αντίστροφη της g και
- η τριάδα $(a^-(t), a^+(t), f(t))$ περιγράφει τις (δεξιά και αριστερά) κλίσεις της f (που είναι γνωστές από τις παραγώγους της g).

Σημειώστε ότι για την $g(t)$ ισχύει το άνω φράγμα που δείξαμε στο Λήμμα 3.2. Θα δείξουμε ότι η πολυπλοκότητα χώρου (δηλαδή το πλήθος των σημείων καμπής) της $\tilde{f}$ είναι $O(\log_{1+\epsilon} \frac{f_{\max}}{f_{\min}})$ που είναι ανεξάρτητο από την πολυπλοκότητα χώρου της ίδιας της f . Όσο αφορά την χρονική πολυπλοκότητα θα δείξουμε ότι είναι $O(\log_{1+\epsilon} \frac{f_{\max}}{f_{\min}} \cdot \log \frac{\tau_N - \tau_0}{\epsilon f_{\min}})$.

Αξίζει να σημειώσουμε ότι αν είχαμε πλήρη γνώση της συνάρτησης f τότε θα μπορούσαμε να χρησιμοποιήσουμε τον αλγόριθμο των Imai και Iri [23] για να πάρουμε μια $(1 + \epsilon)$ -προσέγγιση της f σε χρόνο γραμμικό ως προς την πολυπλοκότητα χώρου της f . Την περίπτωση αυτή ακριβώς περιγράψαμε στην Ενότητα 3.2. Ωστόσο στην περίπτωση που εξετάζουμε τώρα θέλουμε να αποφύγουμε την κατασκευή και την αποθήκευση της f για να την προσεγγίσουμε όσο το δυνατό καλύτερα.



Θα εξετάσουμε ως σημεία παρεμβολής της $\tilde{f}$ τα σημεία τομής της f (που θα είναι το πολύ 2 για κάθε γραμμή) με τις οριζόντιες ευθείες $y_k = (1 + \epsilon)^k \cdot f_{\min}$ για κάθε τιμή $k \in \{0, 1, 2, \dots, k_{\max}\}$. Αρχεί να εξασφαλίσουμε ότι για την δειγματοληψία αυτή ισχύει $(1 + \epsilon)^{k_{\max}} \cdot f_{\min} \leq 2 \cdot f(\frac{\tau_0 + \tau_N}{2})$ έτσι ώστε να καλύψουμε ολόκληρη την συνάρτηση. Δηλαδή η μέθοδος αυτή θα παράγει το πολύ

$$2 \cdot k_{\max} = 2 \cdot \log_{1+\epsilon} \frac{2f((\tau_0 + \tau_N)/2)}{\min\{f(\tau_0), f(\tau_N)\}} \in \mathcal{O} \left(\log_{1+\epsilon} \frac{f_{\max}}{f_{\min}} \right)$$

σημεία παρεμβολής για την $\tilde{f}$.

Φυσικά έχουμε πρόσβαση μόνο στη συνάρτηση $g(t)$ και στην αντίστροφή της $g^{-1}(t)$. Στον Αλγόριθμο 7 δείχνουμε τα βήματα του αλγορίθμου που θα ακολουθήσουμε. Συγκεκριμένα ο αλγόριθμος παράγει σημεία παρεμβολής της $\tilde{f}$ χωρίζοντας την διαδικασία σε δύο φάσεις: Στην πρώτη φάση επεξεργαζόμαστε το πρόθεμα της f έως ότου το πρώτο δειγματοληπτικό σημείο με δεξιά παράγωγο να γίνει το πολύ 1. Τα αντίστοιχα σημεία παρεμβολής δημιουργούνται από διαδοχικές αυξήσεις της συνάρτησης g και στη συνέχεια αναζητήσεις στον χώρο αναζήτησης T . Η δεύτερη φάση επεξεργάζεται το επίθεμα της f με όλες τις παραγωγίσιμες τιμές στο διάστημα $[-1, 1]$. Η περίπτωση αυτή αντιμετωπίζεται πιο εύκολα από μια αναδρομική διχοτόμηση έως ότου τα παραγόμενα σημεία παρεμβολής μας εγγυηθούν τον επιθυμητό λόγο προσέγγισης. Οι αλγόριθμοι 7 και 8 περιγράφουν αναλυτικά την συνολική διαδικασία.

Αρχικοποίηση:

$k \leftarrow 0$

$t_0 \leftarrow \tau_0; g_0 \leftarrow g(\tau_0); \tilde{f}_0 \leftarrow g_0 - \tau_0$

STORE($t_0, \tilde{f}_0$)

Βασικός βρόγχος 1: /* Phase1: slope(g, t) ≥ 2 */

while $a^+(t_k) > 1$ **do**

$k \leftarrow k + 1$

$g_k \leftarrow t_{k-1} + \tilde{f}_{k-1} \cdot (1 + \epsilon')$

$t_k \leftarrow g^{-1}(g_k)$

$\tilde{f}_k \leftarrow g_k - t_k$ /* Προς-τα-πίσω αναζητήσεις */

if $a^-(t_k) \geq 1$ **then**

STORE($t_k, \tilde{f}_k$)

Βασικός βρόγχος 2: /* Phase2: $a^+(t_k) \leq 1$ */

BISECT(t_k, t_N)

Επέστρεψε όλα τα αποθηκευμένα (STORE) σημεία

Αλγόριθμος 7: Σημεία παρεμβολής της $\tilde{f}$ στο $[\tau_0, \tau_N]$



BISECT(c, d)

$l \leftarrow d - c$

$f_c \leftarrow g(c) - c$

$f_d \leftarrow g(d) - d$

$f_{c,d}^{\min} \leftarrow \min\{f_c, f_d\}$

$m \leftarrow \frac{f_d - f_c + ca^+(c) - da^-(d)}{a^+(c) - a^-(d)}$

$error \leftarrow (a^+(c) - a^-(d)) \cdot \frac{(m-c)(d-m)}{l}$

if $error > \epsilon \cdot f_{c,d}^{\min}$ then

$k \leftarrow k + 1$

$t_k \leftarrow \frac{c+d}{2}$

$\tilde{f}_k \leftarrow g(t_k) - t_k$ /* Ευθείες αναζητήσεις */

STORE($t_k, \tilde{f}_k$)

BISECT($c, \frac{c+d}{2}$)

BISECT($\frac{c+d}{2}, d$)

Αλγόριθμος 8: Διχοτόμηση στο $[c, d]$, BISECT(c, d)

Θα αναλύσουμε πρώτα την ορθότητα του αλγορίθμου. Θα αποδείξουμε ότι η γραμμική παρεμβολή της $\tilde{f}$ από τα σημεία παρεμβολής του αλγορίθμου είναι πράγματι μια $(1 + \epsilon)$ -προσέγγιση της f . Συγκεκριμένα θα δείξουμε ότι για κάθε $t \in [\tau_0, \tau_N]$, ισχύει

$$|f(t) - \tilde{f}(t)| \leq \epsilon \cdot f(t) \Leftrightarrow (1 - \epsilon)f(t) \leq \tilde{f}(t) \leq (1 + \epsilon)f(t).$$

Επειδή όλα τα σημεία της $\tilde{f}$ είναι πραγματικά σημεία της f που είναι μια κοίλα συνάρτηση, η $\tilde{f}$ είναι ένα κάτω φράγμα της f , δηλαδή για κάθε $t \in [\tau_0, \tau_N]$ ισχύει,

$$f(t) \geq \tilde{f}(t).$$

Επομένως αρκεί να δείξουμε ότι για κάθε $t \in [\tau_0, \tau_N]$ ισχύει,

$$(1 - \epsilon) \cdot f(t) \leq \tilde{f}(t).$$

Θα αντιμετωπίσουμε τις αντίστοιχες περιπτώσεις κάθε φάσης ξεχωριστά.

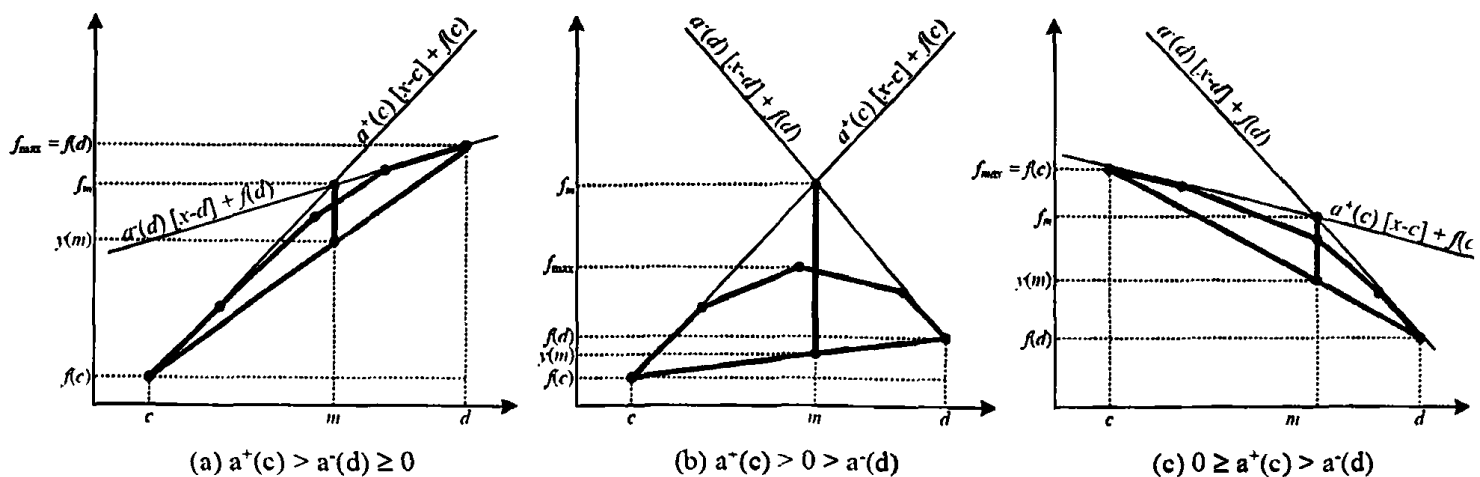
Δύο διαδοχικά σημεία $(t_k, f_k), (t_{k+1}, f_{k+1})$ της Φάσης 1: Ισχύει ότι $f(t_k) = \tilde{f}(t_k) = f_k$ και $f(t_{k+1}) = \tilde{f}(t_{k+1}) = f_{k+1}$. Επίσης για κάθε $t \in [t_k, t_{k+1}]$ ισχύει $f_k \leq \tilde{f}(t) \leq f(t) \leq f_{k+1}$. Επομένως αρκεί να δείξουμε ότι $(1 - \epsilon) \cdot f_{k+1} \leq f_k$. Πράγματι,

$$\begin{aligned} f_{k+1} &= g_{k+1} - t_{k+1} \\ &\leq \dot{g}_{k+1} - t_k = (1 + \epsilon') \cdot f_k \\ \Rightarrow f_k &\geq \frac{f_{k+1}}{(1 + \epsilon')} = (1 - \epsilon) \cdot f_{k+1}, \end{aligned}$$

που ισχύει για $\epsilon' = \frac{\epsilon}{1 - \epsilon}$.

Δύο διαδοχικά σημεία $(t_k, f_k), (t_{k+1}, f_{k+1})$ της Φάσης 2: Θα χρησιμοποιήσουμε το ακόλουθο αποτέλεσμα.





Σχήμα 3.5: Οι τρεις περιπτώσεις για να φράξουμε άνω το απόλυτο σφάλμα μεταξύ δύο διαδοχικών σημείων παρεμβολής. Το άνω φράγμα του απόλυτου σφάλματος παρουσιάζεται από το μπλε ευθύγραμμο τμήμα.

Λήμμα 3.3. Έστω ένα χρονικό διάστημα $[c, d]$ τέτοιο ώστε $a^+(c) > a^-(d)$, με $l = d - c$ και έστω η γραμμική παρεμβολή

$$y(x) = -f(c) \cdot \frac{x-d}{d-c} + f(d) \cdot \frac{x-c}{d-c} = \frac{f(d) - f(c)}{l} \cdot x + \frac{f(c)d - f(d)c}{l}$$

που παρεμβάλλει τα δύο δειγματοληπτικά σημεία $(c, f(c))$ και $(d, f(d))$. Το απόλυτο σφάλμα είναι άνω φραγμένο από

$$\text{MaxAbsError}(c, d) = (a^+(c) - a^-(d)) \cdot \frac{(m-c)(d-m)}{l} \leq \frac{l \cdot (a^+(c) - a^-(d))}{4}$$

όπου

$$m = \frac{f(d) - f(c) + c \cdot a^+(c) - d \cdot a^-(d)}{a^+(c) - a^-(d)}.$$

Απόδειξη. Έστω το τρίγωνο που ορίζεται από τις ακόλουθες τρεις γραμμές (Σχήμα 3.5):

$$\begin{aligned} y(x) &= \frac{f(d) - f(c)}{l} \cdot x + \frac{f(c)d - f(d)c}{l} \\ y_c(x) &= a^+(c) \cdot (x - c) + f(c) \\ y_d(x) &= a^-(d) \cdot (x - d) + f(d) \end{aligned}$$

Η x-συντεταγμένη του σημείου τομής (m, f_m) για τις δύο ευθείες $y_c(x)$ και $y_d(x)$ υπολογίζεται ως εξής:

$$m = \frac{f(d) - f(c) + c \cdot a^+(c) - d \cdot a^-(d)}{a^+(c) - a^-(d)}.$$

Υπενθυμίζουμε ότι η συνάρτηση f είναι κοίλα και τα σημεία $(c, f(c))$ και $(d, f(d))$ είναι πραγματικά σημεία της f . Επομένως ο περιορισμός της f στο διάστημα $[c, d]$ εμφανίζεται εξολοκλήρου στο εμβαδόν του τριγώνου και το μέγιστο δυνατό (απόλυτο) σφάλμα είναι ίσο με την κάθετη απόσταση του τρίτου σημείου (m, f_m) από το ευθύγραμμο τμήμα που ενώνει

τα δύο σημεία $(c, f(c))$ και $(d, f(d))$ (Σχήμα 3.5). Υπολογίζουμε το άνω φράγμα για το απόλυτο σφάλμα. Οι υπολογισμοί είναι ίδιοι και για τις τρεις περιπτώσεις που δείχνουμε στο Σχήμα 3.5. Έστω $a_m = \frac{f(d)-f(c)}{l}$ να είναι η κλίση της ευθείας $y(x)$. Παρατηρήστε ότι

$$\begin{aligned} a_m &= \frac{f(d) - f(c)}{d - c} = \frac{|f_m - f(c)| - |f_m - f(d)|}{l} \\ &= \frac{m - c}{l} \cdot \frac{f_m - f(c)}{m - c} - \frac{d - m}{l} \cdot \frac{f_m - f(d)}{d - m} \\ &= \frac{m - c}{l} \cdot a^+(c) - \frac{d - m}{l} \cdot a^-(d) \end{aligned}$$

Επομένως θα έχουμε,

$$\begin{aligned} \text{MaxAbsError} &= f_m - y(m) = (f_m - f(c)) - (y(m) - f(c)) \\ &= a^+(c) \cdot (m - c) - a_m \cdot (m - c) = (a^+(c) - a_m) \cdot (m - c) \\ &= (a^+(c) - a^-(d)) \cdot \frac{(m - c)(d - m)}{l} \leq \frac{l \cdot (a^+(c) - a^-(d))}{4} \end{aligned}$$

καθώς $(m - c) + (d - m) = d - c = l$ και το γινόμενο $(m - c) \cdot (d - m)$ μεγιστοποιείται στο σημείο $m = \frac{c+d}{2}$. ■

Σημειώνουμε ότι η αναδρομική εκτέλεση της διχοτόμησης εξασφαλίζει ότι τελικά για όλα τα διαστήματα μεταξύ σημείων παρεμβολής της δεύτερης φάσης ισχύει

$$\text{MaxAbsError}(t_k, t_{k+1}) \leq \epsilon \min\{f_k, f_{k+1}\}.$$

Αυτό σημαίνει ότι η $\tilde{f}$ είναι μια $(1 + \epsilon)$ -προσέγγιση της f σε κάθε διάστημα διαδοχικών σημείων παρεμβολής.

Στη συνέχεια εξετάζουμε την πολυπλοκότητα χώρου και χρόνου του αλγορίθμου. Συγκεκριμένα θα εξετάσουμε το πλήθος των σημείων παρεμβολής που διαλέγει τελικά ο αλγόριθμος για τη συνάρτηση $\tilde{f}$ καθώς και το πλήθος των εκτελέσεων για τον υπολογισμό των σημείων αυτών.

Σημεία Παρεμβολής της Φάσης 1: Σημειώνουμε ότι από την μονοτονία της g έχουμε ότι

$$f_{k+1} = g_{k+1} - t_{k+1} \leq g_{k+1} - t_k = f_k \cdot (1 + \epsilon').$$

Επιπρόσθετα καθώς η κλίση της g είναι τουλάχιστον 2 μέχρι το σημείο t_{k+1} γνωρίζουμε ότι

$$g_{k+1} \geq g_k + 2 \cdot (t_{k+1} - t_k) \Rightarrow t_{k+1} - t_k \leq \frac{g_{k+1} - g_k}{2} = \frac{\epsilon'}{2} f_k.$$

Επομένως συμπεραίνουμε τα ακόλουθα:

$$\begin{aligned} f_{k+1} &= g_{k+1} - t_{k+1} = (1 + \epsilon') \cdot f_k - (t_{k+1} - t_k) \\ &\geq (1 + \epsilon') f_k - \frac{\epsilon'}{2} f_k = \left(1 + \frac{\epsilon'}{2}\right) \cdot f_k \\ &\geq \left(1 + \frac{\epsilon'}{2}\right)^{k+1} \cdot f_0 = \left(\frac{2 - \epsilon}{2 - 2\epsilon}\right)^{k+1} \cdot f_0 \end{aligned}$$



καθώς $\epsilon' = \frac{\epsilon}{1-\epsilon}$. Καταλήγουμε δηλαδή ότι το μέγιστο πλήθος των σημείων παρεμβολής της πρώτης φάσης είναι

$$\log_{\frac{2-\epsilon}{2-2\epsilon}} \left(\frac{f_{\max}}{f_{\min}} \right) - 1.$$

Σημεία Παρεμβολής της Φάσης 2: Κάθε διχοτόμηση χωρίζει στη μέση το χρονικό διάστημα που πρόκειται να εξετάσει. Αν το αρχικό διάστημα στο πρώτο κάλεσμα της διαδικασίας BISECT είναι L τότε στο k -στο επίπεδο του δέντρου αναδρομής τα διαστήματα έχουν μήκη $l_k = L/2^k$. Επειδή $0 \leq a^+(c) - a^-(d) \leq 2$, το απόλυτο σφάλμα φράσσεται από $\frac{l_k \cdot (a^+(c) - a^-(d))}{4} \leq \frac{L}{2^{k+1}}$. Επομένως συμπεραίνουμε ότι το ύψος του δέντρου αναδρομής είναι το πολύ $\log_2 \frac{L}{\epsilon \cdot f_{\min}}$. Σημειώνουμε επίσης ότι οι γονείς των φύλλων στο δέντρο αναδρομής αντιστοιχούν στα διαστήματα για τα οποία το απόλυτο σφάλμα είναι μεγαλύτερο από $\epsilon \cdot f_{\min}$. Αυτό σημαίνει ότι στη χειρότερη περίπτωση καμία $(1 + \epsilon)$ -προσέγγιση μπορεί να αποφύγει την τοποθέτηση ενός σημείου παρεμβολής στο αντίστοιχο διάστημα. Έτσι η προτεινόμενη διχοτόμηση χρησιμοποιεί το πολύ διπλάσιο αριθμό σημείων παρεμβολής που απαιτούνται για οποιαδήποτε $(1 + \epsilon)$ -προσέγγιση της συνάρτησης f .

Αξίζει να σημειώσουμε ότι αν γνωρίζαμε εξ αρχής την πλήρη περιγραφή της f θα μπορούσαμε να χρησιμοποιήσουμε μόνο τα σημεία τομής της f με τις παράλληλες ευθείες ως προς τον άξονα του χρόνου για να παράγουμε μια $(1 + \epsilon)$ -προσέγγιση με το πολύ $2 \log_{1+\epsilon} \frac{2f((\tau_0 + \tau_N)/2)}{\min\{f(\tau_0), f(\tau_N)\}}$ σημεία παρεμβολής. Επομένως η μέθοδος διχοτόμησης παράγει το πολύ διπλάσιο αριθμό σημείων παρεμβολής και θα ισχύει,

$$4 \log_{1+\epsilon} \frac{2f((\tau_0 + \tau_N)/2)}{\min\{f(\tau_0), f(\tau_N)\}} \in \mathcal{O} \left(\log_{1+\epsilon} \frac{f_{\max}}{f_{\min}} \right) = \mathcal{O} \left(\frac{1}{\epsilon} \log \frac{f_{\max}}{f_{\min}} \right).$$

Όσο αφορά το κόστος (δηλαδή, το πλήθος των εκτελέσεων από ευθείες αναζητήσεις) υπολογισμού κάθε σημείου παρεμβολής, αυτό θα είναι ίσο με το μήκος του μονοπατιού στο δέντρο αναζήτησης από την ρίζα προς το αντίστοιχο φύλλο για την συγκεκριμένη διχοτόμηση. Επομένως το συνολικό κόστος θα είναι,

$$\mathcal{O} \left(\frac{1}{\epsilon} \log \frac{f_{\max}}{f_{\min}} \cdot \log \frac{L}{\epsilon \cdot f_{\min}} \right)$$

που εκφράζει το πλήθος των ευθειών αναζητήσεων.



ΚΕΦΑΛΑΙΟ 4

ΠΟΛΛΑΠΛΕΣ ΑΦΕΤΗΡΙΕΣ ΣΕ ΧΡΟΝΙΚΑ ΜΕΤΑΒΑΛΛΟΜΕΝΑ ΣΥΝΤΟΜΟΤΕΡΑ ΜΟΝΟΠΑΤΙΑ

4.1 Τεχνικές για Στατικά Συντομότερα Μονοπάτια με Πολλαπλές Αφετηρίες

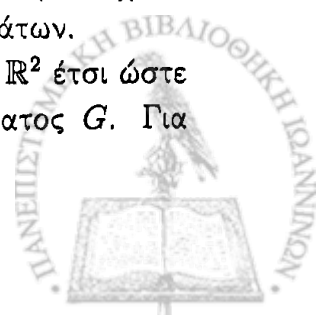
4.2 Επέκταση σε Χρονικά Μεταβαλλόμενα Συντομότερα Μονοπάτια με Πολλαπλές Αφετηρίες

Στο κεφάλαιο αυτό θα μελετήσουμε το πρόβλημα ΣΥΝΤΟΜΟΤΕΡΑ ΜΟΝΟΠΑΤΙΑ ΑΠΟ ΠΟΛΛΑΠΛΕΣ ΑΦΕΤΗΡΙΕΣ – ΣΜΠΑ, στο οποίο σκοπός μας είναι η εύρεση (δέντρων) συντομότερων μονοπατιών από ορισμένες αφετηρίες προς όλες τις υπόλοιπες κορυφές του γραφήματος. Στόχος είναι ο συγκεκριμένος υπολογισμός να κοστίζει ασυμπτωτικά όσο και ο υπολογισμός από μια μόνο αφετηρία. Θα δούμε την στατική εκδοχή του προβλήματος και θα δείξουμε έναν τρόπο που μπορεί να επεκταθεί σε δυναμικό περιβάλλον. Ο στατικός αλγόριθμος στον οποίο θα βασιστούμε επιλύει το πρόβλημα σε επίπεδα γραφήματα όπου οι αφετηρίες βρίσκονται στην εξωτερική όψη του γραφήματος. Στη συνέχεια δείχνουμε έναν αποτελεσματικό αλγόριθμο για να επεκτείνουμε την ιδέα αυτή σε χρονικά μεταβαλλόμενο περιβάλλον χωρίς ωστόσο να περιοριζόμαστε σε επίπεδα γραφήματα και στην συγκεκριμένη τοποθέτηση για τις αφετηρίες.

4.1 Τεχνικές για Στατικά Συντομότερα Μονοπάτια με Πολλαπλές Αφετηρίες

Στη συγκεκριμένη ενότητα θα δείξουμε έναν τρόπο που επιλύει το πρόβλημα για στατικά μονοπάτια με πολλαπλές αφετηρίες σε επίπεδα γραφήματα όπως περιγράφεται στη συνέχεια. Πρώτα ας δώσουμε ορισμένες έννοιες σχετικές με την επιπεδικότητα γραφημάτων.

Ένα γράφημα G ονομάζεται *επίπεδο* αν υπάρχει μια σχεδίαση του G στο $\mathbb{R}^2$ έτσι ώστε δύο οποιεσδήποτε ακμές να έχουν κοινά σημεία μόνο κορυφές του γραφήματος G . Για



ένα επίπεδο γράφημα G και ένα σημείο $x \in \mathbb{R}^2$ καλούμε *όψη* του επίπεδου γραφήματος G το σύνολο των σημείων που μπορούν να συνδεθούν με το σημείο x μέσω μιας καμπύλης Jordan που δεν τέμνει κάποια ακμή του G . Όλες οι όψεις που είναι φραγμένες από ένα σύνολο ακμών ονομάζονται *εσωτερικές όψεις* και η μοναδική μη-φραγμένη όψη καλείται *εξωτερική όψη* που συμβολίζεται με f_∞ . Το *δυϊκό γράφημα* ενός επίπεδου γραφήματος G κατασκευάζεται από τις όψεις του γραφήματος G θεωρώντας αυτές ως κορυφές και οι δυϊκές ακμές ενώνουν δυϊκές κορυφές γειτονικών όψεων του G (όψεις, δηλαδή, του G με κάποια κοινή ακμή). Για μια ακμή e , η αντίστοιχη δυϊκή ακμή έχει κατεύθυνση από την αριστερή όψη της e προς την δεξιά όψη της e θεωρώντας ότι η αριστερή όψη της e βρίσκεται στα αριστερά της e όταν η ακμή e έχει κατεύθυνση προς τα πάνω. Αν προσθέσουμε μια κορυφή $\hat{v}$ στην εξωτερική όψη ενός επίπεδου γραφήματος που ενώνεται με όλες τις κορυφές της εξωτερικής όψης τότε η πρόσθεση της κορυφής δεν πλήττει την επιπεδικότητα. Για μια κορυφή v με εισερχόμενη ακμή (u, v) και εξερχόμενες ακμές (v, w) και (v, x) θα λέμε ότι η (v, x) είναι *αριστερά της* (v, w) ως προς την (u, v) αν η (v, x) εμφανίζεται μεταξύ των (v, w) και (u, v) αριστερόστροφα (αντίθετα ως προς την φορά του ρολογιού). Δοθέντος ενός μονοπατιού P που περιέχει την ακμή (v, w) , θα λέμε ότι η ακμή (v, x) *προέρχεται* αριστερά από το P αν ισχύει τουλάχιστον μια από τις ακόλουθες συνθήκες:

- (a) η ακμή που προηγείται της (v, w) στο μονοπάτι P είναι η (u, v) και η (v, x) είναι αριστερά της (v, w) ως προς την (u, v) .
- (b) η κορυφή v είναι η πρώτη κορυφή του μονοπατιού P και η v βρίσκεται στην εξωτερική όψη και η (v, x) είναι αριστερά της (v, w) ως προς την τεχνητή ακμή $(\hat{v}, v)$.

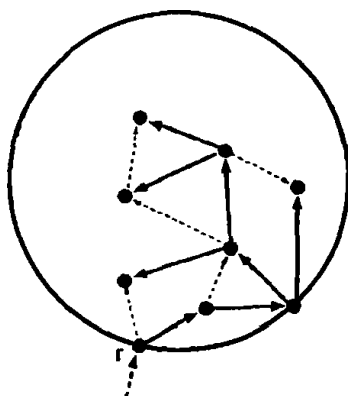
Το πρόβλημα για στατικά μονοπάτια με πολλαπλές αφετηρίες σε επίπεδα γραφήματα περιγράφεται ως εξής:

ΣΥΝΤΟΜΟΤΕΡΑ ΜΟΝΟΠΑΤΙΑ ΑΠΟ ΠΟΛΛΑΠΛΕΣ ΑΦΕΤΗΡΙΕΣ (ΣΜΠΑ)	
<i>Είσοδος:</i>	Ένα επίπεδο γράφημα $G = (V, E)$, $w : E \rightarrow \mathbb{R}^+$, εξωτερική όψη f_∞ και ένα δέντρο συντομότερων μονοπατιών T_0 ριζωμένο σε $r \in f_\infty$
<i>Ερώτημα:</i>	Μια αναπαράσταση για τα δέντρα συντομότερων μονοπατιών ριζωμένα σε όλες τις κορυφές της f_∞

Αξίζει να σημειώσουμε ότι στην είσοδο του προβλήματος μπορούμε να αποφύγουμε το αρχικό δέντρο T_0 και ο υπολογισμός του να γίνει ως αρχικό βήμα στον αλγόριθμο. Ωστόσο για ευκολία θα θεωρήσουμε ότι το δέντρο T_0 μας δίνεται στην είσοδο του προβλήματος. Θα περιγράψουμε έναν αλγόριθμο που επιλύει το παραπάνω πρόβλημα σε $\mathcal{O}(n \log n)$ χρόνο [26]. Σημειώνουμε ότι σε επίπεδα γραφήματα κάθε δέντρο συντομότερων μονοπατιών μπορεί να βρεθεί γραμμικό χρόνο $\mathcal{O}(n)$ [21]. Οπότε η αποτύπωση όλων των δέντρων δεν μπορεί να επιλυθεί σε αποτελεσματικό χρόνο με απλή εκτέλεση του γραμμικού αλγορίθμου. Επιπρόσθετα κάθε δέντρο αποτελείται από $n - 1$ ακμές με αποτέλεσμα η ρητή αναφορά σε κάθε ένα από αυτά να χρειάζεται περισσότερο χρόνο. Για να προσπεράσουμε το πρόβλημα αυτό θα χρησιμοποιήσουμε μια έμμεση αναπαράσταση των δέντρων συντομότερων μονοπατιών.

Έστω $r_0, r_1, \dots, r_s$ να είναι οι κορυφές που ανήκουν στην εξωτερική όψη με δεξιόστροφη διάταξη και έστω $T_0, T_1, \dots, T_s$ τα αντίστοιχα δέντρα συντομότερων μονοπατιών. Στην ουσία ο αλγόριθμος δείχνει τις αλλαγές που χρειάζονται για να μετατρέψει το δέντρο T_0 σε T_1 , και γενικά την μετατροπή κάθε δέντρου T_i σε T_{i+1} . Η χρονική πολυπλοκότητα του αλγορίθμου εξαρτάται από το πλήθος των μετατροπών, οι οποίες θα δείξουμε ότι είναι το πολύ όσες είναι οι κορυφές της εξωτερικής όψης (δηλαδή $O(n)$). Επίσης θα δείξουμε ότι κάθε μετατροπή μπορεί να εκτελεστεί σε $O(\log n)$ χρησιμοποιώντας μια συγκεκριμένη δομή δεδομένων για την αναπαράσταση κάθε δέντρου.

Δοθέντος ενός ριζωμένου δέντρου T και μια κορυφή v του δέντρου, με $T[v]$ συμβολίζουμε το μονοπάτι του δέντρου από τη ρίζα στην κορυφή v . Αν u είναι πρόγονος της v στο δέντρο τότε με $T[u, v]$ συμβολίζουμε το (u, v) -μονοπάτι στο δέντρο T . Θυμίζουμε ότι σε ένα δέντρο συντομότερων μονοπατιών κάθε ακμή είναι relaxed (ικανοποιείται το κριτήριο χαλάρωσης) και ο εκτιμητής απόστασης $D_T(v)$ (μέσω της διαδικασίας χαλάρωσης) περιέχει τις πραγματικές συντομότερες αποστάσεις από την ρίζα.

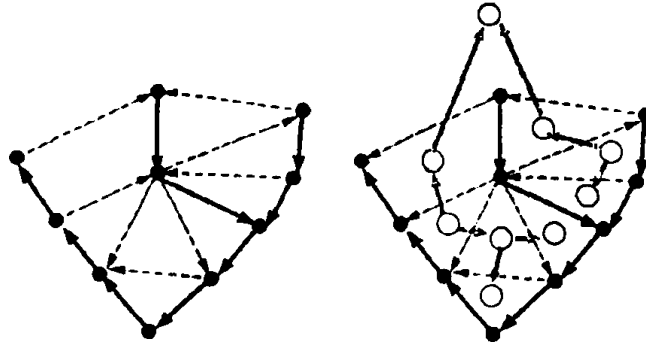


Σχήμα 4.1: Το όριο της εξωτερικής όψης βρίσκεται στον κύκλο. Ένα δεξιότερο δέντρο συντομότερων μονοπατιών ριζωμένο στην κορυφή r φαίνεται με έντονες ακμές.

Μια αναζήτηση κατά βάθος (dfs search) σε ένα επίπεδο γράφημα με τον περιορισμό ότι για κάθε επισκεπτόμενη κορυφή v οι εξερχόμενες ακμές (v, w) ανακαλύπτονται από δεξιά-προς-τα-αριστερά ως προς την ακμή (u, v) (ή ως προς την ακμή $(\hat{u}, v)$ αν η v είναι ρίζα του δέντρου και βρίσκεται στην εξωτερική όψη) από την οποία η κορυφή v πρώτα ανακαλύφθηκε, καλείται *δεξιόστροφη αναζήτηση* (right-first search). Μια δεξιόστροφη αναζήτηση επάγει ένα δέντρο δεξιόστροφης αναζήτησης που αποτελείται από τέτοιες επισκεπτόμενες ακμές, σε αναλογία με το δέντρο αναζήτησης κατά βάθος. Το δεξιότερο δέντρο συντομότερων μονοπατιών ριζωμένο στην κορυφή r_0 αποτελείται από το σύνολο των αποστάσεων $D()$ από την ρίζα r_0 βρίσκοντας ένα δέντρο δεξιόστροφης αναζήτησης στο υπογράφημα που επάγεται από ακμές (u, v) τέτοιες ώστε $D(u) = D(v) + w(u, v)$. Στο Σχήμα 4.1 δείχνουμε ένα τέτοιο δέντρο ριζωμένο στην κορυφή r .

Έστω T το δέντρο συντομότερο μονοπατιών με ρίζα την κορυφή r_0 . Συμβολίζουμε με T^* το σύνολο των ακμών που δεν βρίσκονται στο T . Το δέντρο T αποτελεί ένα σχε-
λετικό δέντρο (δηλαδή ένα υπογράφημα του γραφήματος που περιέχει όλες τις κορυφές και είναι δέντρο) του G . Αντίστοιχα, ως προς το δυϊκό γράφημα του G , το T^* αποτελεί





Σχήμα 4.2: Αριστερά φαίνεται ένα γράφημα και ένα σκελετικό δέντρο με έντονες ακμές. Στα δεξιά δείχνουμε το δυϊκό γράφημα που δεν περιέχει τις ακμές που δεν βρίσκονται στο αρχικό δέντρο.

ένα σκελετικό δέντρο του δυϊκού, θεωρώντας συνεκτικά επίπεδα γραφήματα και αγνοώντας κατευθύνσεις στις ακμές και με ρίζα την κορυφή που αντιπροσωπεύει την εξωτερική όψη. Στο Σχήμα 4.2 δείχνουμε το δέντρο T^* στο δυϊκό γράφημα. Επίσης σημειώστε ότι κάθε ακμή του T^* δεν ικανοποιεί το κριτήριο χαλάρωσης, ειδικά θά είχε συμπεριληφθεί στο δέντρο T σε αντικατάσταση κάποιας άλλης ακμής για να μην δημιουργηθεί κύκλος στο T . Μια ακμή (x, y) καλείται *απόμακρη unrelaxed* (leafmost unrelaxed) αν δεν ικανοποιεί το κριτήριο χαλάρωσης και καμία ακμή που είναι απόγονος της (x, y) δεν ικανοποιεί το κριτήριο χαλάρωσης στο T^* .

Αρχικοποίηση:

for $i = 1$ to s **do**

 πρόσθεσε την ακμή (r_i, r_{i-1}) με $w(r_i, r_{i-1}) = \infty$

$T \leftarrow$ δεξιότερο δέντρο συντομότερων μονοπατιών με ρίζα r_0

Βασικός βρόγχος:

for $i = 1$ to s **do**

 αφαίρεσε την ακμή του δέντρου T που εισέρχεται στην r_i και πρόσθεσε την ακμή (r_i, r_{i-1}) /* τώρα το T είναι ριζωμένο στην κορυφή r_i */

while υπάρχει unrelaxed ακμή **do**

 εφάρμοσε την διαδικασία relax στην απόμακρη unrelaxed ακμή

Αλγόριθμος 9: Στατικά Συντομότερα Μονοπάτια με Πολλαπλές Αφετηρίες

Ο ψευδοκώδικας δίνεται στον Αλγόριθμο 9. Το πρώτο βήμα της πρόσθεσης ακμών $(r_s, r_{s-1}), (r_{s-1}, r_{s-2}), \dots, (r_1, r_0)$ με άπειρο βάρος δεν πλήττει την επιπεδικότητα του γραφήματος καθώς οι κορυφές αυτές βρίσκονται στην εξωτερική όψη του γραφήματος. Στο επόμενο βήμα βρίσκουμε το πρώτο δέντρο με ρίζα την κορυφή r_0 . Το βήμα αυτό μπορεί να επιτευχθεί με την εκτέλεση του στατικού Dijkstra αλγορίθμου τροποποιώντας την επιλογή της επόμενης εξερχόμενης ακμής. Στη συνέχεια τροποποιούμε το τρέχον δέντρο με την εισαγωγή και διαγραφή ακμών προσκείμενες στις δύο αντίστοιχες ρίζες. Το βασικό βήμα

του αλγορίθμου είναι η επιλογή της πιο απόμακρης *unrelaxed* ακμής.

Ο λόγος για τον οποίο επιλέγεται η πιο απόμακρη *unrelaxed* ακμή είναι ο εξής: Έστω $e = (u, v)$ μια ακμή που δεν επιλέγεται στο (σκελετικό) δέντρο συντομότερων μονοπατιών T . Υπάρχει ένα μοναδικό μονοπάτι $P(u, v)$ στο T που ενώνει τα άκρα της ακμής e . Μαζί με την ακμή e το μονοπάτι $P(u, v)$ δημιουργεί έναν κύκλο C_e στο γράφημα G . Οι ακμές που δεν ανήκουν στο δέντρο και οι οποίες βρίσκονται εσωτερικά του κύκλου C_e είναι ακριβώς οι απόγονοι της ακμής e στο δυϊκό σκελετικό δέντρο ριζωμένο στην εξωτερική όψη. Πιο συγκεκριμένα αν η e' είναι μια πιο απόμακρη *unrelaxed* ακμή τότε καμία *unrelaxed* ακμή δε βρίσκεται εσωτερικά του κύκλου $C_{e'}$.

Αποδεικνύεται ότι ο Αλγόριθμος 9 τρέχει σε χρόνο $O(n \log n)$ [26]. Βασίζεται κυρίως στο γεγονός ότι (i) κάθε ακμή γίνεται *relaxed* το πολύ μια φορά και (ii) κάθε επανάληψη του βασικού βρόγχου (δηλαδή κάθε μετατροπή του T_i από το T_{i-1}) υλοποιείται σε $O(\log n)$ χρόνο. Με βάση αυτές τις παρατηρήσεις αποδεικνύεται ότι κάθε ακμή ανήκει σε δέντρα συντομότερων μονοπατιών ριζωμένα σε συνεχόμενες υποακολουθίες του $(r_0, r_1, \dots, r_s)$ [26]. Σημειώνουμε ότι μια από τις βασικές τεχνικές για την δημιουργία των δεξιότερων δέντρων συντομότερων μονοπατιών είναι η πρόσθεση της κορυφής $\hat{v}$ στην εξωτερική όψη του επίπεδου γραφήματος που ενώνεται με όλες τις κορυφές της εξωτερικής όψης. Μια παρόμοια τεχνική θα χρησιμοποιήσουμε σε χρονικά μεταβαλλόμενο περιβάλλον. Επίσης ο αλγόριθμος μπορεί να παράγει μια δομή δεδομένων με $O(n)$ χώρο που για οποιοδήποτε κορυφή s και οποιαδήποτε κορυφή d στην εξωτερική όψη, βρίσκει την πρώτη ακμή στο συντομότερο (s, d) -μονοπάτι σε χρόνο $O(\log \log \text{degree}(d))$. Επομένως μπορεί κανείς χρησιμοποιώντας αυτή την δομή να βρει το συντομότερο (s, d) -μονοπάτι P σε χρόνο $O(|P|)$ αν οι κορυφές του γραφήματος έχουν σταθερό βαθμό.

Μια γενίκευση του αλγορίθμου του Klein βρίσκεται στην πρόσφατη εργασία [3]. Στην πρόσφατη αυτή εργασία εξετάζουν γενικεύσεις επίπεδων γραφημάτων γνωστά ως g -genus γραφήματα, τα οποία μπορούν να σχεδιαστούν σε κάποια σπειροειδή επιφάνεια¹ (torus) χωρίς να τέμνονται οι ακμές. Η χρονική πολυπλοκότητα του αλγορίθμου για την κατασκευή της αντίστοιχης δομής είναι $O(gn \log n)$ [3].

4.2 Επέκταση σε Χρονικά Μεταβαλλόμενα Συντομότερα Μονοπάτια με Πολλαπλές Αφετηρίες

Στην ενότητα αυτή θα μελετήσουμε το πρόβλημα συντομότερων μονοπατιών με πολλαπλές αφετηρίες σε χρονικά μεταβαλλόμενο περιβάλλον. Πιο συγκεκριμένα μας δίνονται ορισμένες κορυφές αφετηρίας $s_1, s_2, \dots, s_q$ και αναζητάμε το συντομότερο χρονικά μεταβαλλόμενο (s_i, v) -μονοπάτι για κάθε χρονική στιγμή $t \in [t_0, t_N]$ και για κάθε κορυφή $v \in V(G)$. Το πρόβλημα αυτό το περιγράφουμε ως εξής:

¹ Η g -σπειροειδή επιφάνεια ορίζεται με βάση την σπείρα (torus) με g οπές. Έτσι η 0-σπειροειδή επιφάνεια είναι η σφαίρα, η 1-σπειροειδή επιφάνεια είναι η σπείρα με μια οπή κ.ο.κ. Ένα g -genus γράφημα είναι εκείνο για το οποίο υπάρχει σχεδίαση χωρίς να τέμνονται οι ακμές για το μικρότερο g , πάνω σε μια g -σπειροειδή επιφάνεια.



ΣΥΝΑΡΤΗΣΕΙΣ ΣΥΝΤΟΜΟΤΕΡΗΣ ΑΦΙΞΗΣ, ΠΡΟΣ ΠΡΟΟΡΙΣΜΟ (ΣΣΠ)

Είσοδος: Ένα γράφημα $G = (V, E)$, γραμμικές συναρτήσεις $A : E \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$, και (κορυφές αφετηρίας) $s_1, s_2, \dots, s_q \in V(G)$

Ερώτημα: (i) Περιγραφή (π.χ., μέσω των σημείων καμπής) των συναρτήσεων $A[s_1, v], A[s_2, v], \dots, A[s_q, v]$ για κάθε $v \in V(G)$ και
(ii) Προσδιορισμός των δέντρων συντομότερων μονοπατιών ριζωμένα στις κορυφές $s_1, s_2, \dots, s_q$

Μια απλή προσέγγιση του προβλήματος είναι η εκτέλεση ενός αλγορίθμου (όπως για παράδειγμα του εξαρτώμενου από την έξοδο αλγορίθμου ή του προσεγγιστικού αλγορίθμου που περιγράψαμε στο Κεφάλαιο 3) q φορές με αφετηρία την κορυφή s_i προς όλες τις υπόλοιπες κορυφές, για κάθε $1 \leq i \leq q$. Κάτι τέτοιο ωστόσο θα επιφέρει q εκτελέσεις του αλγορίθμου. Για παράδειγμα, αν θεωρήσουμε τον αλγόριθμο που είναι χρονικά εξαρτώμενος από την έξοδο (Ενότητα 3.1) τότε θα χρειαστούμε συνολικό χρόνο $O(q \cdot K \cdot \log^2 n)$, όπου K το πλήθος των σημείων καμπής. Ωστόσο, όπως παρουσιάσαμε και στην στατική περίπτωση, μπορεί κανείς να επιτύχει έναν πιο γρήγορο αλγόριθμο για το πρόβλημα αυτό αν καταφέρει να αποθηκεύσει κατάλληλες χρονικές στιγμές κάθε συντομότερο μονοπάτι.

Έστω M η μέγιστη ποσότητα της συνάρτησης καθυστέρησης των συντομότερων (s_i, v) -μονοπατιών για κάθε χρονική στιγμή και για κάθε κορυφή $v \in V(G)$. Δηλαδή θέτουμε για κάθε κορυφή $v \in V(G)$,

$$M = \max_{1 \leq i \leq q} \{D_{s_i, v}(t)\}.$$

Επίσης χωρίζουμε το εύρος των διαστήματος $[t_0, t_N]$ σε $q + 1$ χρονικά διαστήματα:

$$[0, T, 2T, \dots, qT, (q + 1)T].$$

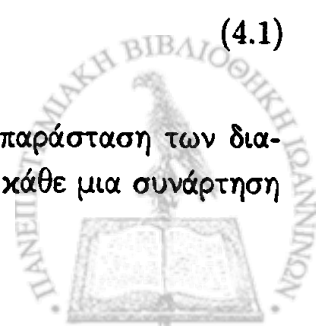
Θα τροποποιήσουμε το γράφημα G και θα κατασκευάσουμε ένα γράφημα G' ως εξής:

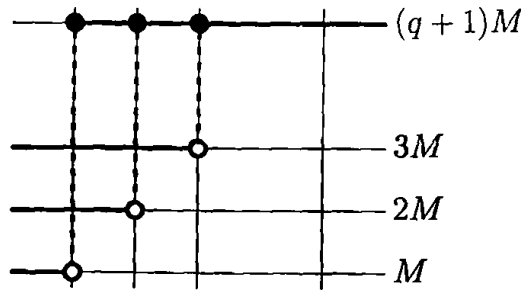
- προσθέτουμε μια νέα κορυφή z και
- προσθέτουμε τις ακμές $(z, s_1), (z, s_2), \dots, (z, s_q)$.

Η κορυφή z στο G' ενώνεται μόνο με τις κορυφές $s_1, s_2, \dots, s_q$ οπότε σε οποιοδήποτε μονοπάτι μεταξύ των κορυφών z και v η δεύτερη κορυφή του μονοπατιού θα είναι μια από τις q κορυφές αναχώρησης. Οι συναρτήσεις καθυστέρησης των ακμών του γραφήματος G παραμένουν αμετάβλητες στο G' ενώ για τις q νέες ακμές που προσθέσαμε στο G' , ορίζουμε την ακόλουθη τμηματική συνάρτηση:

$$D_{z, s_i}(t) = \begin{cases} i \cdot M & 0 \leq t < i \cdot T \\ (q + 1) \cdot M & t \geq i \cdot T \end{cases}, \quad (4.1)$$

για κάθε $1 \leq i \leq q$. Στο Σχήμα 4.3 δείχνουμε μια σχηματική αναπαράσταση των διαστημάτων της συνάρτησης καθυστέρησης. Αξίζει να σημειώσουμε ότι κάθε μια συνάρτηση $D_{z, s_i}(t)$ περιέχει ένα σημείο καμπής από τον παραπάνω ορισμό.





Σχήμα 4.3: Μια σχηματική αναπαράσταση της συνάρτησης καθυστέρησης $D_{z,s_i}(t)$.

Στο γράφημα G' θα αναζητήσουμε το χρονικά μεταβαλλόμενο συντομότερο (z, v) -μονοπάτι, για κάθε κορυφή $v \in V(G)$. Για κάθε χρονική περίοδο $(i-1)T \leq t < iT$ όπου $1 \leq i \leq q$ προσεγγιστικά θέλουμε να υπολογίσουμε την συνάρτηση συντομότερης άφιξης $A[z, v]$ από την κορυφή z μέσω της κορυφής s_i προς τον προορισμό v . Αν λοιπόν το (z, v) -μονοπάτι για την συγκεκριμένη χρονική περίοδο ξεκινάει με την ακμή (z, s_i) τότε πράγματι η αναπαράσταση κάθε (s_i, v) -μονοπατιού γίνεται με σωστό τρόπο. Αυτό ακριβώς επιτυγχάνουμε με το ακόλουθο λήμμα.

Λήμμα 4.1. Στο γράφημα G' κάθε (z, v) -μονοπάτι την χρονική περίοδο $(i-1)T \leq t < iT$ περιέχει την ακμή (z, s_i) .

Απόδειξη. Επειδή η κορυφή z ενώνεται μόνο με τις κορυφές αναχώρησης $s_1, s_2, \dots, s_q$, η πρώτη ακμή στο μονοπάτι $P_{z,v}$ είναι της μορφής (z, s_1) ή (z, s_2) ή $\dots$ ή (z, s_q) . Αρκεί λοιπόν να δείξουμε ότι την συγκεκριμένη χρονική περίοδο διαλέγει την ακμή (z, s_i) . Έστω ότι το μονοπάτι $P_{z,v}$ περιέχει την ακμή (z, s_j) με $j \neq i$ και $1 \leq j \leq q$.

Συμβολίζουμε με P'_j το συντομότερο (z, v) -μονοπάτι μέσω της ακμής (z, s_j) , δηλαδή $P'_j = (z, s_j) \cup P_j$ με P_j να είναι το συντομότερο (s_j, v) -μονοπάτι. Ανάλογα δηλώνουμε το συντομότερο (z, v) -μονοπάτι P'_i μέσω της ακμής (z, s_i) , δηλαδή $P'_i = (z, s_i) \cup P_i$ με P_i να είναι το συντομότερο (s_i, v) -μονοπάτι. Σύμφωνα με την αρχική μας θεώρηση θα πρέπει να ισχύει $D[P'_j](t) < D[P'_i](t)$ έτσι ώστε το (z, v) -μονοπάτι να διαλέξει την ακμή (z, s_j) . Θα έχουμε,

$$D[P'_j](t) < D[P'_i](t) \Leftrightarrow D_{z,s_j}(t) + D[P_j](t) < D_{z,s_i}(t) + D[P_i](t). \quad (4.2)$$

Από τον ορισμό που δώσαμε για τις συναρτήσεις $D_{z,s_i}(t)$ (Εξίσωση 4.1), την χρονική περίοδο $(i-1)T \leq t \leq iT$ θα ισχύει,

$$D_{z,s_j}(t) = \begin{cases} (q+1) \cdot M & j < i \\ j \cdot M & j > i \end{cases}. \quad (4.3)$$

Αντίστοιχα από τον ορισμό της Εξίσωσης 4.1, για την ακμή (z, s_i) θα έχουμε,

$$D_{z,s_i}(t) = i \cdot M \quad (4.4)$$

Θα κοιτάξουμε ξεχωριστά τις περιπτώσεις $j < i$ και $i < j$.



- $j < i$: Από την Εξίσωση 4.3 γνωρίζουμε ότι $D_{z,s_j}(t) = (q+1) \cdot M$. Αντικαθιστώντας $D_{z,s_i}(t) = i \cdot M$ στην Εξίσωση 4.2 έχουμε,

$$(q+1) \cdot M + D[P_j](t) < i \cdot M + D[P_i](t) \Leftrightarrow$$

$$q+1 + \frac{D[P_j](t)}{M} < i + \frac{D[P_i](t)}{M}.$$

Και επειδή $M = \max_{1 \leq i \leq q} \{D_{s_i,v}(t)\}$ θα ισχύει $0 \leq \frac{D[P_i](t)}{M} \leq 1$ για κάθε $1 \leq i \leq q$. Δηλαδή,

$$q+1 + \frac{D[P_j](t)}{M} < i + \frac{D[P_i](t)}{M} \Leftrightarrow$$

$$q+1 < q+1 + \frac{D[P_j](t)}{M} < i + \frac{D[P_i](t)}{M} < i+1.$$

Επομένως $q < i$ και καταλήγουμε σε άτοπο καθώς $i \leq q$.

- $i < j$: Αντίστοιχα από την Εξίσωση 4.3 θα έχουμε ότι $D_{z,s_j}(t) = j \cdot M$ και η Εξίσωση 4.2 γίνεται τώρα,

$$j \cdot M + D[P_j](t) < i \cdot M + D[P_i](t) \Leftrightarrow$$

$$j + \frac{D[P_j](t)}{M} < i + \frac{D[P_i](t)}{M} \Leftrightarrow$$

$$j < j + \frac{D[P_j](t)}{M} < i + \frac{D[P_i](t)}{M} < i+1.$$

Δηλαδή $j \leq i$ και καταλήγουμε πάλι σε άτοπο καθώς στην περίπτωση αυτή θα πρέπει $i < j$.

Επομένως σε κάθε περίπτωση καταλήγουμε σε άτοπο και η αρχική μας θεώρηση ότι το μονοπάτι $P_{z,v}$ περιέχει άλλη ακμή εκτός της (z, s_i) δεν μπορεί να ισχύει. Δηλαδή η πρώτη ακμή του μονοπατιού $P_{z,v}$ την χρονική περίοδο $(i-1)T \leq t < iT$ είναι η ακμή (z, s_i) όπως περιγράφεται στο λήμμα. ■

Επομένως πράγματι αναπαριστάμε τα συντομότερα (s_i, v) -μονοπάτια στο γράφημα G μέσω της αναζήτησης των χρονικά μεταβαλλόμενων (z, v) -μονοπατιών στο γράφημα G' .

Όσο αφορά την πολυπλοκότητα της αναζήτησης, θα υπολογίσουμε το πλήθος των σημείων καμπής στο γράφημα G' . Έχουμε ένα σημείο καμπής από κάθε συνάρτηση $D_{z,s_i}(t)$ για τις q ακμές που προσθέσαμε στην κορυφή z . Δηλαδή συνολικά θα έχουμε $K + q$ σημεία καμπής για το (z, v) -μονοπάτι όπου K είναι το πλήθος των σημείων καμπής στο γράφημα G μεταξύ των (s_i, v) -μονοπατιών. Επομένως αν εφαρμόσουμε τον αλγόριθμο που είναι χρονικά εξαρτώμενος από την έξοδο (Ενότητα 3.1) τότε θα χρειαστούμε χρόνο $\mathcal{O}((q+K) \cdot \log^2 n)$, που είναι γρηγορότερο από την απλή εκτέλεση του ίδιου αλγορίθμου q φορές.



ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ

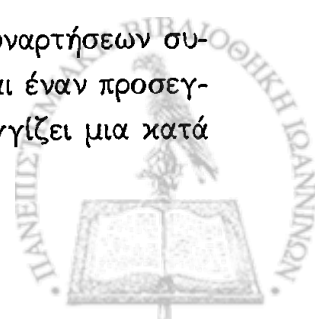
5.1 Συμπεράσματα και Επεκτάσεις

5.1 Συμπεράσματα και Επεκτάσεις

Στην εργασία αυτή μελετήσαμε το πρόβλημα της εύρεσης του χρονικά μεταβαλλόμενου συντομότερου μονοπατιού. Παρουσιάσαμε δύο παραλλαγές του προβλήματος όπου η μια από αυτές στοχεύει επιπρόσθετα στην προσέγγιση ελάχιστων τιμών που περιγράφονται από άγνωστες συναρτήσεις. Δείξαμε έναν τρόπο που μεταφέρει έναν κλασικό αλγόριθμο από το στατικό περιβάλλον σε χρονικά μεταβαλλόμενο όταν υποθέτουμε την FIFO ιδιότητα σε ένα δίκτυο. Συσχετίσαμε το πρόβλημα του χρονικά μεταβαλλόμενου συντομότερου μονοπατιού με τα παραμετροποιημένα συντομότερα μονοπάτια όταν οι συναρτήσεις του χρόνου περιγράφονται από κατά τμήματα γραμμικές συναρτήσεις. Μέσα από αυτό το πρόβλημα αλλά και από προηγούμενα αποτελέσματα δείξαμε γνωστά άνω και κάτω υπολογιστικά φράγματα ως προς το πλήθος των σημείων καμπής σε ένα χρονικά μεταβαλλόμενο δίκτυο.

Μια από τις βασικές μας θεωρήσεις ήταν η περιγραφή του χρόνου μετάβασης από κατά τμήματα γραμμικές συναρτήσεις. Στη συγκεκριμένη περίπτωση δείξαμε ότι τα παραμετροποιημένα συντομότερα μονοπάτια συμπεριφέρονται με παρόμοιο τρόπο με τα χρονικά μεταβαλλόμενα συντομότερα μονοπάτια. Θα μπορούσε κανείς να επεκτείνει την συνάρτηση χρόνου από γραμμικές συναρτήσεις σε πολυώνυμα βαθμού d . Ωστόσο σε αυτή την περίπτωση η συμπεριφορά των δύο προβλημάτων διαφέρει. Για παράδειγμα στα παραμετροποιημένα συντομότερα μονοπάτια η πολυπλοκότητα ως προς το κάτω και άνω φράγμα παραμένει ένα πολυώνυμο βαθμού d ενώ σε ένα χρονικά μεταβαλλόμενο δίκτυο μπορεί να φτάσει έως και d^n . Αλλά ακόμα και σε αυτή την περίπτωση παραμένει ενδιαφέρον η αναζήτηση και η τεκμηρίωση αντίστοιχων φραγμάτων.

Παρουσιάσαμε επίσης αλγοριθμικά αποτελέσματα για υπολογισμούς συναρτήσεων συντομότερης άφιξης: έναν αλγόριθμο χρονικά εξαρτώμενο από την έξοδο και έναν προσεγγιστικό αλγόριθμο. Σημειώσαμε ότι ο προσεγγιστικός αλγόριθμος προσεγγίζει μια κατά



τμήματα γραμμική συνάρτηση σε ένα χρονικά μεταβαλλόμενο δίκτυο. Αναλύσαμε περαιτέρω τον προσεγγιστικό αλγόριθμο έτσι ώστε να προσεγγίσει άγνωστη συνεχή κοίλα συνάρτηση με ελάχιστη κλίση ≥ -1 για την οποία υποθέσαμε ότι μας δίνονται ορισμένες τιμές της άγνωστης συνάρτησης μαζί με τις αντίστοιχες κλίσεις της συνάρτησης. Αν επιπρόσθετα είχαμε γνώση της κατά τμήματα γραμμικής συνάρτησης τότε για την περίπτωση αυτή δείξαμε επίσης έναν γραμμικό αλγόριθμο που προσεγγίζει την συνάρτηση αυτή.

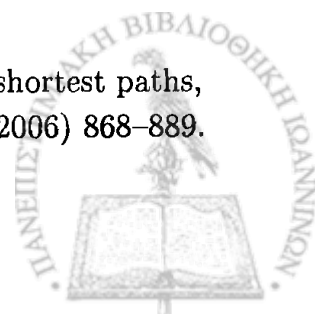
Ως προς τα αλγοριθμικά αποτελέσματα που παρουσιάσαμε για υπολογισμούς συναρτήσεων συντομότερης άφιξης είναι ενδιαφέρον να αναζητήσει κανείς κλάσεις γραφημάτων όπου η πολυπλοκότητα των συναρτήσεων συντομότερης άφιξης να παραμένει πολυωνυμική. Για παράδειγμα στην κλάση των επίπεδων ή g -genus γραφημάτων είναι άγνωστο αν το πλήθος των σημείων καμπής μπορεί να φτάσει στα φράγματα που δείξαμε.

Επιπρόσθετα παρουσιάσαμε έναν γνωστό αποτελεσματικό αλγόριθμο για την εύρεση συντομότερων μονοπατιών από όλες τις κορυφές της εξωτερικής όψης ενός επίπεδου γραφήματος προς όλους τους δυνατούς προορισμούς. Εχμεταλλεuthήκαμε ορισμένες ιδιότητες του αλγορίθμου και παρουσιάσαμε ένα αποτελεσματικό αλγόριθμο για το πρόβλημα με πολλαπλές αφετηρίες σε ένα χρονικά μεταβαλλόμενο περιβάλλον χωρίς ωστόσο να περιοριζόμαστε σε επίπεδα γραφήματα. Τέλος, μια κατεύθυνση που μπορεί κανείς να κοιτάξει είναι μια τυχαιοποιημένη προσέγγιση του προβλήματος. Για παράδειγμα δεν γνωρίζουμε αν μια μικρή τυχαία διαταραχή των συναρτήσεων χρόνου κατά μήκος μιας ακμής μπορεί να οδηγήσει σε μια σχετικά μικρή διαταραχή της αναμενόμενης πολυπλοκότητας των συναρτήσεων συντομότερης άφιξης.



ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, 1993.
- [2] R. K. Ahuja, J. B. Orlin, S. Pallottino, and M. Scutell'a, Dynamic shortest paths minimizing travel times and costs, The shortest route through a network with time-dependent internodal transit times, *Networks* **41** (2003) 197–205.
- [3] S. Cabello, E. W. Chambers, and J. Erickson, Multiple-Source Shortest Paths in Embedded Graphs, Technical Report, CoRR abs/1202.0314, 2012.
- [4] I. Chabini and S. Lan, Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks., *IEEE Transactions on Intelligent Transportation Systems* **3** (2002) 60–74.
- [5] L. Cooke and E. Halsey, The shortest route through a network with time-dependent internodal transit times, *Journal of Mathematical Analysis and Applications* **14** (1966) 492–498.
- [6] B. C. Dean, Shortest paths in fifo time-dependent networks: theory and algorithms, Technical Report, Massachusetts Institute Of Technology, 2004.
- [7] D. Delling, Time-dependent SHARC-routing, *Algorithmica* **60** (2011) 60–94.
- [8] D. Delling and D. Wagner, Landmark-Based Routing in Dynamic Graphs, *Proc. WEA 2007*, (2007) 52–65.
- [9] E. W. Dijkstra, A note on two problems in connexion with graphs, *Numerische Mathematik* **1** (1959) 269–271.
- [10] D.H. Douglas and T.K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, *Canadian Cartographer* **10** (1973) 112–122.
- [11] S. Dreyfus, An appraisal of some shortest-path algorithms, *Operations Research* **17** (1969) 395–412.
- [12] J. Fakcharoenphol and S. Rao, Planar graphs, negative weight edges, shortest paths, and near linear time, *Journal of Computer and System Sciences* **72** (2006) 868–889.



- [13] M. L. Fredman and R. E. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms, *Journal of the ACM* **34** (1987) 596–615.
- [14] L. R. Ford and D. R. Fulkerson, *Modern Heuristic Techniques for Combinatorial Problems*, Princeton University Press, Princeton, NJ, 1962.
- [15] L. Foschini, J. Hershberger, and S. Suri, On the Complexity of Time-Dependent Shortest Paths, *Proc. SODA 2011*, (2011) 327–341.
- [16] R. Geisberger, P. Sanders, D. Schultes, and D. Delling., Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks, *Proc. WEA 2008*, (2008) 319–333.
- [17] A. Goldberg and C. Harrelson, Computing the Shortest Path: A* Meets Graph Theory, *Proc. SODA 2005*, (2005) 156–165.
- [18] A. Goldberg, H. Kaplan, and R. Werneck. , Reach for A*: Efficient point-to-point shortest path algorithms, *Proc. ALENEX 2006*, (2006) 129–143.
- [19] D. Gusfield, Parametric combinatorial computing and a problem of program module distribution, *Journal of the ACM* **30** (1983) 551–563.
- [20] E. Hart, N. Nilsson, and B. Raphael, A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on Systems, Science and Cybernetics* **4** (1968) 100–107.
- [21] M. Henzinger, P. Klein, S. Rao, and S. Subramanian, Faster shortest-path algorithms for planar graphs, *Journal of Computer and System Sciences* **55** (1997) 3–23.
- [22] T. Ikeda, M. Tsu, H. Imai, S. Nishimura, H. Shimoura, T. Hashimoto, K. Tenmoku, and K. Mitoh, A Fast Algorithm for Finding Better Routes by AI Search Techniques, *Proc. IEEE Vehicle Navigation and Information Systems 2004*, (2004) 291–296.
- [23] H. Imai and M. Iri, An optimal algorithm for approximating a piecewise linear function, *Journal of Information Processing* **9** (1986) 159–162.
- [24] R. M. Karp and J. B. Orlin, Parametric shortest path algorithms with an application to cyclic staffing, *Discrete Applied Mathematics* **3** (1981) 37–45.
- [25] D. E. Kaufman and R. L. Smith, Fastest paths in time-dependent networks for intelligent vehicle-highway systems application, *Journal of Intelligent Transportation Systems* **1** (1993) 1–11.
- [26] P. N. Klein, Multiple-source Shortest Paths in Planar Graphs, *Proc. SODA 2005*, (2005) 146–155.
- [27] K. Mulmuley and P. Shah, A lower bound for the shortest path problem, *Journal of Computer and System Sciences* **63** (2001) 253–267.



- [28] G. Nannicini, *Point-to-Point Shortest Paths on Dynamic Time-Dependent Road Networks*, PhD Thesis, Ecole Polytechnique, 2009.
- [29] S. Neubauer, *Space Efficient Approximation of Piecewise Linear Functions*, M.Sc. Thesis, Universitat Karlsruhe, 2010.
- [30] T. A. J. Nicholson, Finding the shortest route between two points in a network, *Computer J.* **9** (1966) 275–280.
- [31] A. Orda and R. Rom, Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length, *Journal of the ACM* **37** (1990) 607–625.
- [32] A. Orda and R. Rom, Minimum weight paths in time-dependent networks, *Networks* **21** (1991) 295–319.
- [33] P. Sanders and D. Schultes, Dynamic highway-node routing, *Proc. WEA 2007*, (2007) 66–79.
- [34] S. Suri, A linear time algorithm with minimum link paths inside a simple polygon, *Computing Vision Graphics Image Processing* **35** (1986) 99–110.
- [35] G. T. Toussaint and D. Avis, On a convex hull algorithm for polygons and its application to triangulation problems, *Pattern Recognition* **15** (1982) 23–29.
- [36] I. Tomek, Two algorithms for piecewise-linear continuous approximation of functions of one variable, *IEEE Transactions on Computing* **C-23** (1974) 445–448.
- [37] N. E. Young, R. E. Tarjan, and J. B. Orlin, Faster parametric shortest path and minimum balance algorithms, *Networks* **21** (1991) 1–17.
- [38] A. Ziliaskopoulos and H. Mahmassani, A time-dependent shortest path algorithm for real-time intelligent vehicle highway system applications, *Transportation Research Record* **1408** (1993) 94–104.



ΣΥΝΤΟΜΟ ΒΙΟΓΡΑΦΙΚΟ

Η Χρυσούλα Τσιρώνη γεννήθηκε το 1979 στα Ιωάννινα. Το έτος 2002 εισήχθη στο Τμήμα Πληροφορικής του Ελληνικού Ανοικτού Πανεπιστημίου από το οποίο αποφοίτησε το 2009. Τον Φεβρουάριο του 2010 εισήχθη στο Μεταπτυχιακό Πρόγραμμα Σπουδών του Τμήματος Πληροφορικής του Πανεπιστημίου Ιωαννίνων όπου ολοκλήρωσε τις υποχρεώσεις της με την παρούσα διατριβή τον Φεβρουάριο του 2013.

