

ΑΠΟΤΕΛΕΣΜΑΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ & ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΣΤΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ

Διδακτορική Διατριβή

ΔΗΜΗΤΡΗΣ Ι. ΑΝΤΩΝΙΟΥ

Επιβλέπων: ΧΡΗΣΤΟΣ ΜΑΚΡΗΣ, ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

Ιανουάριος 2011

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής

“In a world we need to share, it's not enough to stand and stare.”

ΑΠΟΤΕΛΕΣΜΑΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ
ΚΑΙ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΜΕ
ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΑΝΑΚΤΗΣΗ
ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΣΤΙΣ
ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Δημήτρης Ι. Αντωνίου

Επταμελής Επιτροπή

Χρήστος Μακρής, (Επιβλέπων)

Αθανάσιος Τσακαλίδης, Καθηγητής (Μέλος Τριμελούς)

Ιωάννης Γαροφαλάκης, Καθηγητής (Μέλος Τριμελούς)

Γεώργιος Παυλίδης, Καθηγητής

Σπυρίδων Λυκοθανάσης, Καθηγητής

Ιωάννης Χατζηλυγερούδης, Επίκουρος Καθηγητής

Κωνσταντίνος Πουλάς, Επίκουρος Καθηγητής

Σχολή: Πολυτεχνική

Τμήμα: Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής

Τομέας: Λογικού των Υπολογιστών

Στους γονείς μου, στη Δέσποινα και στο Γιάννη

*‘Remember when you were young,
You shone like the sun.
Shine on you crazy diamond.
Now there's a look in your eyes,
Like black holes in the sky.
Shine on you crazy diamond.
You were caught on the crossfire
Of childhood and stardom,
Blown on the steel breeze.
Come on you target for faraway laughter,
Come on you stranger, you legend, you martyr, and shine!’*

Πίνακας Περιεχομένων

ΣΧΕΤΙΚΗ ΒΙΒΛΙΟΓΡΑΦΙΑ	20
ΔΙΕΘΝΗ ΕΠΙΣΤΗΜΟΝΙΚΑ ΠΕΡΙΟΔΙΚΑ	23
ΔΙΕΘΝΗ ΕΠΙΣΤΗΜΟΝΙΚΑ ΣΥΝΕΔΡΙΑ.....	23
ΚΕΦΑΛΑΙΟ 1.....	27
ΕΙΣΑΓΩΓΙΚΕΣ ΈΝΝΟΙΕΣ.....	28
1.1 ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ	28
1.2 ΣΥΜΠΙΕΣΗ ΔΕΔΟΜΕΝΩΝ	30
1.3 ΕΠΑΝΟΡΓΑΝΩΣΗ ΔΙΚΤΥΑΚΩΝ ΤΟΠΩΝ	31
1.4 ΣΥΣΤΗΜΑΤΑ ΣΥΣΤΑΣΕΩΝ ΚΑΙ ΕΞΑΤΟΜΙΚΕΥΣΗΣ ΙΣΤΟΤΟΠΩΝ ..	33
1.5 ΑΞΙΟΛΟΓΗΣΗ ΠΟΙΟΤΗΤΑΣ ΠΗΓΑΙΟΥ ΚΩΔΙΚΑ	35
1.5.1 ΤΟ ΠΡΟΤΥΠΟ ISO9126.....	35
1.5.2 ΕΚΤΙΜΗΣΗ ΤΗΣ ΣΥΝΤΗΡΗΣΙΜΟΤΗΤΑΣ ΛΟΓΙΣΜΙΚΟΥ	38
ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΕΦΑΛΑΙΟΥ	38
ΚΕΦΑΛΑΙΟ 2.....	42
ΕΦΑΡΜΟΓΗ ΤΩΝ SPLAY ΔΕΝΤΡΩΝ ΣΤΗ ΣΥΜΠΙΕΣΗ ΔΕΔΟΜΕΝΩΝ.....	43
2.1 SPLAY ΔΕΝΤΡΑ.....	44
2.2 ΠΩΣ ΕΦΑΡΜΟΖΟΥΜΕ SPLAY ΓΙΑ LOGLOGN- ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ	51
2.2.1 ΕΦΑΡΜΟΖΟΝΤΑΣ SPLAYING ΓΙΑ ΝΑ ΕΠΙΤΥΧΟΥΜΕ ΤΟ ΧΑΜΗΛΟΤΕΡΟ ΟΡΙΟ ΜΕΣΑ ΣΕ ΕΝΑΝ ΠΑΡΑΓΟΝΤΑ $O(\log\log N)$	51
2.2.2 ΣΥΝΘΗΚΗ ΣΕΒΑΣΜΟΥ ΑΛΥΣΙΔΑΣ	52

2.2.3	CHAIN SPLAY ΜΕ ΠΟΛΥΠΛΟΚΟΤΗΤΑ ΧΕΙΡΟΤΕΡΗΣ ΠΕΡΙΠΤΩΣΗΣ O(LOGN) 53	
2.2.4	ΤΥΧΑΙΟΠΟΙΗΜΕΝΟ CHAIN SPLAY	55
2.3	ΕΦΑΡΜΟΓΗ ΤΩΝ SPLAY ΔΕΝΤΡΩΝ ΣΤΗ ΣΥΜΠΙΕΣΗ ΔΕΔΟΜΕΝΩΝ 56	
2.4	Ο ΑΛΓΟΡΙΘΜΟΣ SPLAY-ΠΡΟΘΕΜΑΤΟΣ.....	57
2.5	SPLAY ΣΥΜΠΙΕΖΟΝΤΑΣ ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΤΗΝ ΤΕΧΝΙΚΗ ΤΟΥ CHAIN 58	
2.6	ΠΕΙΡΑΜΑΤΙΚΗ ΣΥΓΚΡΙΣΗ ΤΩΝ ΔΙΑΦΟΡΕΤΙΚΩΝ ΜΕΘΟΔΩΝ ΣΥΜΠΙΕΣΗΣ. 59	
2.7	ΣΥΓΚΡΙΣΗ ΤΩΝ ΤΥΧΑΙΟΠΟΙΗΜΕΝΩΝ ΚΑΙ ΜΗ ΤΥΧΑΙΟΠΟΙΗΜΕΝΩΝ ΕΚΔΟΣΕΩΝ	60
	BIBΛΙΟΓΡΑΦΙΑ ΚΕΦΑΛΑΙΟΥ	65
	ΚΕΦΑΛΑΙΟ 3.....	69
	ΑΝΑΘΕΣΗ HOTLINKS ΒΑΣΙΖΟΜΕΝΗ ΣΤΗΝ ΟΜΟΙΟΤΗΤΑ ΠΕΡΙΕΧΟΜΕΝΟΥ	70
3.1	ΕΙΣΑΓΩΓΗ.....	70
3.2	ΠΡΟΗΓΟΥΜΕΝΕΣ ΕΡΓΑΣΙΕΣ	72
3.3	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΠΙΣΚΟΠΗΣΗ	74
3.4	HOTLINKS ΜΕΤΡΩΝΤΑΣ ΤΗΝ ΑΠΟΔΟΣΗ ΕΝΟΣ ΑΛΓΟΡΙΘΜΟΥ ΑΝΑΘΕΣΗΣ 75	
3.4.1	ΤΟ ΜΟΝΤΕΛΟ	75
3.4.2	ΟΙ ΜΕΘΟΔΟΙ ΑΞΙΟΛΟΓΗΣΗΣ.....	78
3.4.2.1	Μετρική απόστασης.....	78
3.4.2.2	Μετρική Βάρους.....	78
3.4.2.3	Μετρική με Βάση την Εντροπία	79
3.4.2.4	Μετρική με Βάση το Μήκος Μονοπατιού	79
3.5	Ο ΑΛΓΟΡΙΘΜΟΣ.....	80
3.6	ΠΕΙΡΑΜΑΤΑ	81
3.7	ΑΞΙΟΛΟΓΗΣΗ.....	84
3.8	ΣΥΝΕΙΣΦΟΡΑ ΤΗΣ ΕΡΓΑΣΙΑΣ & ΣΥΜΠΕΡΑΣΜΑΤΑ	90

ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΕΦΑΛΑΙΟΥ	90
ΚΕΦΑΛΑΙΟ 4.....	95
ΜΙΑ ΤΕΧΝΙΚΗ ΕΞΑΤΟΜΙΚΕΥΣΗΣ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΠΡΟΧΩΡΗΜΕΝΕΣ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ: Η ΠΕΡΙΠΤΩΣΗ ΤΩΝ ΕΞΑΡΣΕΩΝ ΕΠΙΣΚΕΥΕΩΝ ΣΕ ΙΣΤΟΣΕΛΙΔΕΣ	96
4.1 ΕΙΣΑΓΩΓΗ.....	96
4.2 ΠΕΡΙΓΡΑΦΗ ΕΝΟΣ ΣΕΝΑΡΙΟΥ BURST.	98
4.3 ΚΑΤΗΓΟΡΙΕΣ ΚΑΙ ΙΣΤΟΣΕΛΙΔΕΣ	98
4.4 Η ΘΕΩΡΙΑ ΤΗΣ ΕΞΑΤΟΜΙΚΕΥΣΗΣ ΚΑΙ ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ	100
4.5 ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΕΞΑΤΟΜΙΚΕΥΣΗ	101
4.5.1 SPLAY ΔΕΝΤΡΟ	102
4.5.2 ΔΥΑΔΙΚΟΣ ΣΩΡΟΣ (BINARY HEAP)	102
4.6 Η ΠΡΟΤΕΙΝΟΜΕΝΗ ΛΥΣΗ: ΕΞΑΤΟΜΙΚΕΥΣΗ ΣΤΗΝ ΠΕΡΙΠΤΩΣΗ ΣΕΝΑΡΙΟΥ ΕΞΑΡΣΕΩΝ ΕΠΙΣΚΕΥΕΩΝ	103
4.7 Η ΑΝΑΛΥΣΗ ΧΩΡΟΥ ΚΑΙ ΧΡΟΝΟΥ	107
4.7.1 Η ΛΥΣΗ ΜΕ ΠΙΝΑΚΕΣ.....	107
4.7.2 Η ΠΡΟΤΕΙΝΟΜΕΝΗ ΛΥΣΗ	108
4.7.2.1 Απαιτήσεις σε Χώρο	108
4.7.2.2 Απαιτήσεις σε Χρόνο.....	108
4.8 ΠΕΙΡΑΜΑΤΑ	108
4.8.1 ΤΟ ΣΥΝΟΛΟ ΔΕΔΟΜΕΝΩΝ «4 ΠΑΝΕΠΙΣΤΗΜΙΑ»	109
4.8.2 ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΠΕΙΡΑΜΑΤΩΝ	109
4.8.3 ΑΠΟΔΟΤΙΚΟΤΗΤΑ ΑΠΟ ΤΗΝ ΑΠΟΨΗ ΤΟΥ ΧΡΟΝΟΥ ΑΠΟΚΡΙΣΗΣ 111	
4.8.4 ΑΠΟΔΟΤΙΚΟΤΗΤΑ ΩΣ ΠΡΟΣ ΤΟ ΧΡΟΝΟ ΜΕ ΤΗ ΧΡΗΣΗ ΠΟΛΛΑΠΛΩΝ ΜΟΤΙΒΩΝ ΕΞΑΡΣΕΩΝ ΚΑΙ ΔΙΑΦΟΡΕΤΙΚΟ ΑΡΙΘΜΟ ΧΡΗΣΤΩΝ ΚΑΙ ΙΣΤΟΣΕΛΙΔΩΝ	112
4.9 ΕΦΑΡΜΟΓΗ ΣΗΜΑΣΙΟΛΟΓΙΚΗΣ ΕΚΔΟΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ΣΕ ΟΝΤΟΛΟΓΙΕΣ.....	118
4.10 ΑΠΟΔΟΤΙΚΗ ΑΝΕΥΡΕΣΗ ΕΞΑΡΣΕΩΝ ΕΠΙΣΚΕΥΕΩΝ	119

4.10.1	ΑΠΟΔΟΤΙΚΗ ΑΝΕΥΡΕΣΗ ΕΞΑΡΣΕΩΝ ΕΠΙΣΚΕΨΕΩΝ	120
4.10.2	ΑΝΑΛΥΣΗ ΧΩΡΟΥ ΚΑΙ ΧΡΟΝΟΥ	122
4.10.3	ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΠΕΙΡΑΜΑΤΩΝ	123
ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΕΦΑΛΑΙΟΥ		127
ΚΕΦΑΛΑΙΟ 5.....		132
ΕΦΑΡΜΟΓΗ ΣΥΣΤΑΔΟΠΟΙΗΣΗΣ ΓΙΑ ΤΗΝ ΣΥΝΔΡΟΜΗ ΤΗΣ ΑΞΙΟΛΟΓΗΣΗΣ ΤΗΣ ΣΥΝΤΗΡΗΣΙΜΟΤΗΤΑΣ ΤΟΥ ΠΗΓΑΙΟΥ ΚΩΔΙΚΑ ΣΥΜΦΩΝΑ ΜΕ ΤΟ ΠΡΟΤΥΠΟ ISO/IEC-9126.....		133
5.1	ΕΙΣΑΓΩΓΗ.....	133
5.2	ΧΡΗΣΗ ΜΕΤΡΙΚΩΝ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΤΗ ΒΕΛΤΙΩΣΗ ΤΗΣ ΣΥΝΤΗΡΗΣΙΜΟΤΗΤΑΣ.....	134
5.3	Η ΠΡΟΤΕΙΝΟΜΕΝΗ ΜΕΘΟΔΟΛΟΓΙΑ ΟΜΑΔΟΠΟΙΗΣΗΣ ΜΕ ΣΚΟΠΟ ΤΗΝ ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ ΣΥΝΤΗΡΗΣΙΜΟΤΗΤΑ ΛΟΓΙΣΜΙΚΟΥ	136
5.4	ΕΦΑΡΜΟΓΕΣ - ΑΞΙΟΛΟΓΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ	139
5.5	ΜΕΛΛΟΝΤΙΚΕΣ ΕΡΓΑΣΙΕΣ	142
ΒΙΒΛΙΟΓΡΑΦΙΑ ΚΕΦΑΛΑΙΟΥ		142

Λίστα Σχημάτων

Εικόνα 1 Η διάσπαση της ποιότητας σε χαρακτηριστικά και υπο-χαρακτηριστικά σύμφωνα με το πρότυπο ISO9126.....	36
Εικόνα 2 Το βήμα Splaying. Ο κόμβος που προσπελάζεται είναι ο x . (α) Zig, (b) zig-zig, (c) zig-zag.	45
Εικόνα 3 Splaying στον κόμβο a	46
Εικόνα 4 Ειδικές περιπτώσεις Splaying. (a) Zig-zig μόνο και η τερματική. (b) Zig-zag μόνο και η τερματική.	46
Εικόνα 5 Το Splaying μετά από ανεπιτυχή προσπέλαση του στοιχείου 80	47
Εικόνα 6 Υλοποίηση της join και split: (a) join(t_1, t_2) (b) split(i, t).	48
Εικόνα 7 Υλοποίηση της insertion και delete με χρήση των join και split: (a) insert(i, t) (b) delete(i, t)	49
Εικόνα 8 Εναλλακτική υλοποίηση της insertion και delete. Εισαγωγή του 80 και στη συνέχεια διαγραφή του 30.	50
Εικόνα 9 Η LeftGroup πράξη	53
Εικόνα 10 Δεξιά Περιστροφή	54
Εικόνα 11 Μία απλή περιστροφή με τις πράξεις concatenate και split	55
Εικόνα 12 Semi-splaying στον αριστερότερο κόμβο-φύλλο σε ένα δένδρο κώδικα	57
Εικόνα 13 Συμπίεση μέσω splay δέντρων της λέξης "aabg".	58
Εικόνα 14 Απόδοση σε χρόνο και περιστροφές που εκτελούνται από τον τυχαιοποιημένο αλγόριθμο για πιθανότητες 100%, 50%, 25% and 12.5%. Κάθε χρόνο αντιπροσωπεύει μία διαφορετική ακολουθία προσβάσεων.	61
Εικόνα 15 Σύγκριση διαφορετικών τεχνικών συμπίεσης σε διάφορα σύνολα δεδομένων	62
Εικόνα 16 Η συμπίεση που επιτυγχάνεται από κάθε τεχνική συμπίεσης σε κάθε βιολογική ακολουθία. Το μέγεθος μετριέται σε bytes.	64
Εικόνα 17 Αρχιτεκτονική επισκόπηση της εφαρμογής του προτεινόμενου αλγόριθμου σε ένα website.	74
Εικόνα 18 Offline συλλογή και προετοιμασία των δεδομένων - Είσοδος στον αλγόριθμο ανάθεσης hotlinks.	75
Εικόνα 19 Στατιστικά δεδομένα επισκεψιμότητας του www.ceid.upatras.gr	84

Εικόνα 20 Ποσοστό κόμβων που βελτιώσαν τη Simetric μετά από την εφαρμογή του αλγορίθμου	85
Εικόνα 21 Βελτιώσεις που επιτυγχάνονται στα μήκη μονοπατιών.....	85
Εικόνα 22 Ποσοστό κόμβων που βελτιώσαν τη μετρική απόστασης.....	86
Εικόνα 23 Ποσοστό της βελτίωσης σε σχέση με τη μετρική απόστασης.	86
Εικόνα 24 Μέση μείωση της απόστασης από την αρχική σελίδα των, μετρημένη σε συνδέσμους.....	87
Εικόνα 25 Ποσοστό αύξησης της εντροπίας ενός ιστότοπου μετά την εφαρμογή του αλγορίθμου (πυκνοί γράφοι).....	88
Εικόνα 26 Ποσοστό αύξησης της εντροπίας ενός ιστότοπου μετά την εφαρμογή του αλγορίθμου (λιγότερο πυκνοί γράφοι).....	88
Εικόνα 27 Ebay.com :Κατηγοριοποιημένη αναζήτηση	99
Εικόνα 28 Google.com: Το Διαδίκτυο οργανωμένο ανά θέμα σε κατηγορίες.....	99
Εικόνα 29 Παράδειγμα Δυναδικής Σωρού	103
Εικόνα 30 Διάγραμμα ροής προτεινόμενου αλγορίθμου	104
Εικόνα 31 Παράδειγμα εκτέλεσης του αλγορίθμου στο επίπεδο των δομών δεδομένων	106
Εικόνα 32 Ποσοστό Ανάκλησης με βάση τα αποτελέσματα του προτεινόμενου αλγορίθμου	111
Εικόνα 33 Συνολικός χρόνος εκτέλεσης σε χιλιοστά του δευτερολέπτου για κάθε αλγόριθμο μετά από κάθε πρόσβαση.....	112
Εικόνα 34 Χρόνος που απαιτείται για ορισμένες ακολουθίες πρόσβασης, αλλά για διαφορετικά μεγέθη εξάρσεων (σε χιλιοστά του δευτερολέπτου).	114
Εικόνα 35 Μέσος όρος, μέγιστος και ελάχιστος χρόνος που απαιτείται για κάθε πρόσβαση για εξάρσεις μεγεθών 1,2,3,4,5,6 και 7 επισκέψεων.	115
Εικόνα 36 Ο χρόνος που απαιτείται για ένα σταθερό αριθμό 100 χρηστών που εκτελούν προσβάσεις σε σύνολα ιστοσελίδων μεγέθους 500, 1000, 5000 και 10,000.	117
Εικόνα 37 Ο χρόνος που απαιτείται για κάθε πρόσβαση στην περίπτωση των 10.000 ιστοσελίδων και πολλαπλούς αριθμούς χρηστών	118
Εικόνα 38 Τα τρία κύρια στάδια του αλγορίθμου σύστασης.....	120
Εικόνα 39 Εφαρμόζοντας Splaying στην πιο πρόσφατη bursty ιστοσελίδα προς τη ρίζα του splay δέντρου	121
Εικόνα 40 Τα timestamps χρησιμοποιούνται για να διασφαλιστεί ότι ο αριθμός των προσβάσεων και το χρονικό διάστημα στο οποίο λαμβάνουν χώρα συνιστούν μια έξαρση επισκέψεων	122
Εικόνα 41 Αριθμός Bursts για μέγεθος εξάρσης 4000 hits για το χρονικό διάστημα μιας ημέρας.....	123

Εικόνα 42 Αριθμός Bursts για μέγεθος έξαρσης 2000 hits για το χρονικό διάστημα μιας ημέρας.....	124
Εικόνα 43 Αριθμός Bursts για μέγεθος έξαρσης 1000 hits για το χρονικό διάστημα μιας ημέρας.....	124
Εικόνα 44 Εμφανίσεις ανά ημέρα για την πιο δημοφιλή σελίδα	125
Εικόνα 45 Αρχιτεκτονική Συστήματος Αξιολόγησης Συντηρησιμότητας Πηγαίου Κώδικα	138
Εικόνα 46 Συστάδες Συντηρησιμότητας του Apache Geronimo σύμφωνα με το ISO 9126	140

Λίστα Πινάκων

Πίνακας 1.....	63
Πίνακας 2.....	63
Πίνακας 3.....	65
Πίνακας 4 Περίληψη του logfile του www.ceid.upatras.gr	83
Πίνακας 5 Σύγκριση προτεινόμενου αλγορίθμου με άλλες τεχνικές.....	89
Πίνακας 6 Απαιτούμενος χρόνος ανά πρόσβαση για διαφορετικά μεγέθη εξάρσεων	116
Πίνακας 7 Απαιτούμενος χρόνος ανά πρόσβαση για διαφορετικούς αριθμούς ιστοσελίδων	117
Πίνακας 8 Απαιτούμενος χρόνος ανά πρόσβαση για διαφορετικούς αριθμούς χρηστών	118
Πίνακας 9 Αριθμός ελάρσεων για την πιο δημοφιλή σελίδα για διαφορετικά μεγέθη εξάρσεων και το χρονικό διάστημα των 24 ωρών	125
Πίνακας 10 Αριθμός ελάρσεων για την πιο δημοφιλή σελίδα για διαφορετικά μεγέθη εξάρσεων και το χρονικό διάστημα μίας ώρας	125
Πίνακας 11 Παράδειγμα εκτέλεσης του αλγορίθμου για μία ημέρα και για μέγεθος εξάρσης 2000 επισκέψεις.....	127
Πίνακας 12 Στατιστικά Συστάδων.....	140
Πίνακας 13 Μετρικές Συστάδας 2	141

Ευχαριστίες

Καταρχήν, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Χρήστο Μακρή, για τη μακροχρόνια συνεργασία μας, για τη σημαντική βοήθεια που μου παρείχε και για την πολύτιμη καθοδήγηση που μου προσφέρει μέχρι και σήμερα.

Επιπλέον, θα ήθελα να ευχαριστήσω θερμά τον καθηγητή κ. Αθανάσιο Τσακαλίδη, για την πολύτιμη βοήθειά του στα πλαίσια της εκπόνησης της διδακτορικής μου διατριβής, αλλά και στα πλαίσια της ευρύτερης συνεργασίας μας όλα αυτά τα χρόνια.

Ευχαριστώ επίσης τον καθηγητή κ. Γιάννη Γαροφαλάκη, για την καθοριστική συμβολή του ως μέλος της τριμελούς επιτροπής της διδακτορικής διατριβής.

Θα ήθελα να ευχαριστήσω θερμά τους καθηγητές, κ. Σπυρίδων Λυκοθανάση και κ. Γεώργιο Παυλίδη, τον επίκουρο καθηγητή κ. Κωνσταντίνο Πουλά και τον επίκουρο καθηγητή κ. Ιωάννη Χατζηλυγερούδη, για την πρόθυμη συμμετοχή τους στην επταμελή επιτροπή της διδακτορικής μου διατριβής.

Επίσης, θα ήθελα να ευχαριστήσω θερμά το φίλο και συνεργάτη Δρ. Γιάννη Τζήμα, ο οποίος βοήθησε καθοριστικά να πάρω κρίσιμες αποφάσεις, αλλά και να σχεδιάσω το μέλλον μου.

Τέλος, θα ήθελα να ευχαριστήσω μέσα από την καρδιά μου τους γονείς μου για την απεριόριστη υποστήριξη, υπομονή και κατανόηση που επέδειξαν σε όλο το διάστημα των σπουδών μου. Ειλικρινά δε θα είχα καταφέρει τίποτα χωρίς αυτούς.

Δημήτρης Αντωνίου,
Ιανουάριος 2011

Επιτελική Σύνοψη

Αντικείμενο της παρούσας διδακτορικής διατριβής είναι η μελέτη και τροποποίηση βασικών δομών δεδομένων με σκοπό τη δημιουργία νέων και την τροποποίηση υπαρχουσών λύσεων, με εφαρμογές στην Ανάκτηση Πληροφορίας, τη Βιοπληροφορική και το Διαδίκτυο.

Αρχικά, δίνεται έμφαση στην ανάπτυξη και πειραματική επιβεβαίωση αλγοριθμικών τεχνικών για τη σχεδίαση αυτοοργανώμενων δομών δεδομένων (self-organizing data structures). Μέχρι σήμερα, ο μόνος πιθανός υποψήφιος αλγόριθμος αναζήτησης σε δένδρο που μπορεί να είναι $O(1)$ -ανταγωνιστικός είναι το splay δένδρο (splay tree) που παρουσιάστηκε από τους Sleator και Tarjan [1]. Επιπρόσθετα, μελετώνται διάφορες εναλλακτικές τεχνικές αυτοοργάνωσης ([2],[3],[4],[5],[6]) και γίνεται επιβεβαίωση των πάνω ορίων που ισχύουν για την απόδοση των splay trees και για αυτές. Η ανάπτυξη των διάφορων αλγοριθμικών αυτών τεχνικών βρίσκει εφαρμογές πάνω στη συμπίεση δεδομένων. Οι αλγόριθμοι συμπίεσης δεδομένων μπορούν να βελτιώσουν την αποδοτικότητα με την οποία τα δεδομένα αποθηκεύονται ή μεταφέρονται, μέσω της μείωσης του ποσού της πλεονάζουσας πληροφορίας. Η χρήση αυτών των αλγορίθμων τόσο στην κρυπτογράφηση όσο και στην επεξεργασία εικόνας είναι αποδοτική και έχει μεγάλο ερευνητικό ενδιαφέρον. Γενικότερα, οι αυτοοργανώμενες δομές δεδομένων χρίζουν ιδιαίτερης προσοχής στους on-line αλγόριθμους. Αναλυτικότερα, στην παρούσα διατριβή, εφαρμόζεται συμπίεση σε βιολογικά δεδομένα αλλά και σε κείμενα τόσο με χρήση του κλασσικού splay δέντρου [10] αλλά και της $\log \log n$ ανταγωνιστικής παραλλαγής του. Επιπλέον, παρουσιάζονται τυχαιοποιημένες εκδόσεις των παραπάνω δομών και εφαρμόζονται και αυτές στη συμπίεση δεδομένων. Οι $\log \log n$ ανταγωνιστικές δομές έχουν καλύτερη απόδοση όσον αφορά την πολυπλοκότητά τους σε σχέση με την κλασσική splay δομή. Το γεγονός αυτό επιβεβαιώνεται πειραματικά, όπου η επιτυγχανόμενη συμπίεση είναι στις περισσότερες των περιπτώσεων καλύτερη από την αντίστοιχη της κλασσικής δομής.

Επιπλέον, ιδιαίτερο ερευνητικό ενδιαφέρον βρίσκει η εφαρμογή βασικών δομών δεδομένων στο διαδίκτυο. Επιδιώκουμε την ανάπτυξη και θεωρητική επιβεβαίωση αλγορίθμων για προβλήματα όπως η ανάθεση «καυτών συνδέσμων» (hot links [7]), η αναδιοργάνωση ιστοσελίδων και η ανάκτηση πληροφορίας ([8],[9]). Σε πρώτο στάδιο, προτείνονται ευριστικοί αλγόριθμοι με σκοπό την ανάθεση «καυτών συνδέσμων» (hotlinks) και τη βελτίωση της τοπολογίας ενός ιστότοπου ([12],[13],[14]). Σκοπός του αλγορίθμου είναι η προώθηση των δημοφιλών ιστοσελίδων ενός ιστότοπου, μέσω της ανάθεσης συνδέσμων προς αυτές, από ιστοσελίδες οι οποίες είναι σχετικές με αυτές ως προς το περιεχόμενο αλλά και ταυτόχρονα συντελούν στη μείωση της απόστασής τους από την αρχική σελίδα. Παρουσιάζεται το μοντέλο του αλγορίθμου, καθώς και μετρικές

οι οποίες χρησιμοποιούνται για την ποσοτική αξιολόγηση της αποδοτικότητας του αλγορίθμου σε σχέση με ειδικά χαρακτηριστικά ενός ιστότοπου, όπως η εντροπία του.

Σε δεύτερο στάδιο, γίνεται μελέτη τεχνικών προσωποποίησης ιστοσελίδων [11]. Συγκεκριμένα, σκοπός είναι η υλοποίηση ενός αλγορίθμου, ο οποίος θα ανακαλύπτει την αυξημένη ζήτηση μίας κατηγορίας ιστοσελίδων Α από έναν χρήστη και αξιοποιώντας την καταγεγραμμένη συμπεριφορά άλλων χρηστών, θα προτείνει κατηγορίες σελίδων οι οποίες προτιμήθηκαν από χρήστες οι οποίοι ομοίως παρουσίασαν αυξημένο ενδιαφέρον προς την κατηγορία αυτή. Αναλύεται το φαινόμενο της έξαρσης επισκεψιμότητας (burst) και η αξιοποίηση του στο πεδίο της εξατομίκευσης ιστοσελίδων. Ο αλγόριθμος υλοποιείται με τη χρήση δύο δομών δεδομένων, των Binary heaps και των Splay δέντρων, και αναλύεται η χρονική και χωρική πολυπλοκότητά του. Επιπρόσθετα, γίνεται πειραματική επιβεβαίωση της ορθής και αποδοτικής εκτέλεσης του αλγορίθμου. Αξίζει να σημειωθεί πως ο προτεινόμενος αλγόριθμος λόγω της φύσης του, χρησιμοποιεί χώρο, ο οποίος επιτρέπει τη χρησιμοποίηση του στη RAM. Τέλος, ο προτεινόμενος αλγόριθμος δύναται να βρει εφαρμογή σε εξατομίκευση σελίδων με βάση το σημασιολογικό τους περιεχόμενο σε αντιστοιχία με το διαχωρισμό τους σε κατηγορίες.

Σε τρίτο στάδιο, γίνεται παρουσίαση πρωτότυπης τεχνικής σύστασης ιστοσελίδων [15] με χρήση Splay δέντρων. Σε αυτή την περίπτωση, δίνεται ιδιαίτερο βάρος στην εύρεση των σελίδων που παρουσιάζουν έξαρση επισκεψιμότητας και στη σύστασή τους στους χρήστες ενός ιστότοπου. Αρχικά, τεκμηριώνεται η αξία της εύρεσης μιας σελίδας, η οποία δέχεται ένα burst επισκέψεων. Η έξαρση επισκεψιμότητας (burst) ορίζεται σε σχέση τόσο με τον αριθμό των επισκέψεων, όσο και με το χρονικό διάστημά επιτέλεσής τους. Η εύρεση των σελίδων επιτυγχάνεται με τη μοντελοποίηση ενός ιστότοπου μέσω ενός splay δέντρου. Με την τροποποίηση του δέντρου μέσω της χρήσης χρονοσφραγίδων (timestamps), ο αλγόριθμος είναι σε θέση να επιστρέφει σε κάθε χρονική στιγμή την ιστοσελίδα που έχει δεχθεί το πιο πρόσφατο burst επισκέψεων. Ο αλγόριθμος αναλύεται όσον αφορά τη χωρική και χρονική του πολυπλοκότητα και συγκρίνεται με εναλλακτικές λύσεις. Μείζονος σημασίας είναι η δυνατότητα εφαρμογής του αλγορίθμου και σε άλλα φαινόμενα της καθημερινότητας μέσω της ανάλογης μοντελοποίησης. Παραδείγματος χάριν, στην περίπτωση της απεικόνισης ενός συγκοινωνιακού δικτύου μέσω ενός γράφου, ο αλγόριθμος σύστασης δύναται να επιστρέφει σε κάθε περίπτωση τον κυκλοφοριακό κόμβο ο οποίος παρουσιάζει την πιο πρόσφατη συμφόρηση.

Τέλος, όσον αφορά το πεδίο της ανάκτησης πληροφορίας, η διατριβή επικεντρώνεται σε μία πρωτότυπη και ολοκληρωμένη μεθοδολογία με σκοπό την αξιολόγηση της ποιότητας ενός συστήματος λογισμικού βάσει του Προτύπου Ποιότητας ISO/IEC-9126.

Το κύριο χαρακτηριστικό της είναι ότι ολοκληρώνει την αξιολόγηση ενός συστήματος λογισμικού ενσωματώνοντας την αποτίμηση όχι μόνο των χαρακτηριστικών που είναι προσανατολισμένα στο χρήστη, αλλά και εκείνων που είναι πιο τεχνικά και αφορούν τους μηχανικούς λογισμικού ενός συστήματος. Σε αυτή τη διατριβή δίνεται βάρος στην εφαρμογή μεθόδων εξόρυξης δεδομένων πάνω στα αποτελέσματα της μέτρησης μετρικών οι οποίες συνθέτουν τα χαρακτηριστικά του πηγαίου κώδικα, όπως αυτά ορίζονται από το Πρότυπο Ποιότητας ISO/IEC-9126 [16][17]. Ειδικότερα εφαρμόζονται αλγόριθμοι συσταδοποίησης με σκοπό την εύρεση τμημάτων κώδικα με ιδιαίτερα χαρακτηριστικά, που χρήζουν προσοχής.

Σχετική Βιβλιογραφία

- [1] SELF-ADJUSTING BINARY SEARCH TREES ,DANIEL DOMINIC SLEATOR ,ROBERT ENDRE TARJAN, JOURNAL OF THE ACM, VOLUME 32 , ISSUE 3 (JULY 1985)PAGES: 652 - 686 .
- [2] ALTERNATIVES TO SPLAY TREES WITH $O(\log n)$ WORST-CASE ACCESS TIME, J. IACONO ,IN SYMPOSIUM ON DISCRETE ALGORITHMS, PP. 516-522, 2001
- [3] A SIMPLIFIED AND DYNAMIC UNIFIED STRUCTURE, MIHAI B,ADOIU AND ERIK D. DEMAINE.
- [4] HOW TO SPLAY FOR $\log \log n$ – COMPETITIVENESS, GEORGE F. GEORGAKOPOULOS, WEA 2005: 570-579
- [5] DYNAMIC OPTIMALITY—ALMOST, ERIK D. DEMAINE, DION HARMON, JOHN IACONO, MIHAI PATRASCU, FOCS 2004: 484-490.
- [6]DYNAMIC OPTIMALITY AND MULTI-SPLAY TREES,DANIEL DOMINIC SLEATOR AND CHENGWEN CHRIS WANG,NOVEMBER 5, 2004,CMU-CS-04-171.
- [7] TOWARDS ADAPTIVE WEB SITES: CONCEPTUAL FRAMEWORK AND CASE STUDY,M. PERKOWITZ AND O. ETZIONI. COMPUTER NETWORKS, 31(11-16):1245–1258, 1999.
- [8] MANAGING GIGABYTES: COMPRESSING AND INDEXING DOCUMENTS AND IMAGES (IAN H. WITTEN, ALISTAIR MOFFAT, AND TIMOTHY C. BELL)
- [9] MODERN INFORMATION RETRIEVAL (RICARDO BAEZA-YATES AND BERTHIER RIBEIRO-NETO)
- [10] APPLICATION OF SPLAY TREES TO DATA COMPRESSION, D. W. JONES, IN COMMUNICATIONS OF THE ACM, AUGUST 1988, PP. 996-1007.
- [11] WEBSITE PERSONALIZATION BASED ON LINK ANALYSIS AND NAVIGATIONAL PATTERNS.EIRINAKI,M.,VAZIRGIANNIS,M.,2007. ACM TRANSACTIONS ON INTERNET TECHNOLOGY7(4).
- [12] EVALUATION OF HOTLINK ASSIGNMENT HEURISTICS FOR IMPROVING WEB ACCESS, J. CZYZOWICZ, E. KRANAKIS, D. KRIZANC, A. PELC, M. VARGAS MARTIN, IN: PROCEEDINGS OFTHE SECOND INTERNATIONAL CONFERENCE ON INTERNET COMPUTING (IC'2001), JUNE 25TH–28TH, 2001, LAS VEGAS, NV, USA, PP. 793–799.
- [13] DYNAMIC HOTLINKS, K. DOUIEB, S. LANGERMAN, IN: PROCEEDINGS OF THE WORKSHOP ON ALGORITHMS AND DATA STRUCTURES (WADS 2005), LNCS, VOL. 3608, 2005,PP. 271–280.
- [14] NEAR-ENTROPY HOTLINK ASSIGNMENTS, K. DOUIEB, S. LANGERMAN, IN: PROCEEDINGS OF EUROPEAN SYMPOSIUM ON ALGORITHMS (ESA06), LNCS, VOL. 4618, 2006, PP.292–303.
- [15] INTRODUCTION TO RECOMMENDER SYSTEMS: ALGORITHMS AND EVALUATION, JOSEPH A. KONSTAN, ACM TRANS. INF. SYST., VOL. 22, NO. 1. (JANUARY 2004), PP. 1-4.
- [16] K-ATTRACTORS, A CLUSTERING ALGORITHM FOR SOFTWARE MEASUREMENT DATA ANALYSIS, Y. KANELLOPOULOS, P. ANTONELLIS, C. TJORTJIS, C. MAKRIS, IN PROCEEDINGS OF IEEE 19TH INTERNATIONAL CONFERENCE ON TOOLS FOR ARTIFICIAL INTELLIGENCE
- [17] INTERPRETATION OF SOURCE CODE CLUSTERS IN TERMS OF ISO/IEC-9126 QUALITY ASPECTS, KANELLOPOULOS Y., HEITLAGER I., TJORTJIS C., VISSER J., IN THE PROCEEDINGS OF IEEE 12TH CONFERENCE OF SOFTWARE MAINTENANCE AND REENGINEERING (CSMR 2008).

Executive Summary

In this dissertation we take an in-depth look at the use of effective and efficient data structures and algorithms in the fields of data mining and web technologies. The main goal is to develop algorithms based on appropriate data structures, in order to improve the performance at all levels of web applications.

In the first chapter the reader is introduced to the main issues studied dissertation. In the second chapter, we propose novel randomized versions of the splay trees. We have evaluated the practical performance of these structures in comparison with the original version of splay trees and with their log log n-competitive variations, in the application field of compression. Moreover, we show that the Chain Splay tree achieves $O(\log n)$ worst-case cost per query. In order to evaluate performance, we utilize plain splay trees, the log log n-competitive variations, the proposed randomized version with the Chain Splay technique to compress data. It is observed experimentally that the compression achieved in the case of the log log n-competitive technique is, as expected, more efficient than the one of the plain splay trees.

The third chapter focuses on hotlinks assignment techniques. Enhancing web browsing experience is an open issue frequently dealt using hotlinks assignment between webpages, shortcuts from one node to another. Our aim is to provide a novel, more efficient approach to minimize the expected number of steps needed to reach expected pages when browsing a website. We present a randomized algorithm, which combines the popularity of the webpages, the website structure, and for the first time to the best authors' knowledge, the similarity of context between pages in order to suggest the placement of suitable hotlinks. We verify experimentally that users need less page transitions to reach expected information pages when browsing a website, enhanced using the proposed algorithm.

In the fourth chapter we investigate the problem of web personalization. The explosive growth in the size and use of the World Wide Web continuously creates new great challenges and needs. The need for predicting the users' preferences in order to expedite and improve the browsing though a site can be achieved through personalizing of the Websites. Recommendation and personalization algorithms aim at suggesting WebPages to users based on their current visit and past users' navigational patterns. The problem that we address is the case where few WebPages become very popular for short periods of time and are accessed very frequently in a limited temporal space. Our aim is to deal with these bursts of visits and suggest these highly accessed pages to the future users that have common interests. Hence, in this paper, we propose a new web personalization technique, based on advanced data structures. The data structures that are used are the Splay tree (1) and Binary heaps (2). We describe the architecture of the technique, analyze the time and space complexity and prove its performance. In addition, we compare both theoretically and experimentally the proposed technique to another

approach to verify its efficiency. Our solution achieves $O(P^2)$ space complexity and runs in $k \log P$ time, where k is the number of pages and P the number of categories of WebPages.

Extending this algorithm, we propose an algorithm which efficiently detects bursts of visits to webpages. As an increasing number of Web sites consist of multiple pages, it is more difficult for the visitors to rapidly reach their own target. This results in an urgent need for intelligent systems that effectively support the users' navigation to high demand Web content. In many cases, due to specific conditions, web pages become very popular and receive excessively large number of hits. Therefore, there is a high probability that these web pages will be of interest to the majority of the visitors at a given time. The data structure that is used for the purposes of the recommendation algorithm is the Splay tree. We describe the architecture of the technique, analyze the time and space complexity and show its performance.

The dissertation's last chapter elaborates on how to use clustering for the evaluation of a software system's maintainability according to the ISO/IEC-9126 quality standard. More specifically it proposes a methodology that combines clustering and multicriteria decision aid techniques for knowledge acquisition by integrating groups of data from source code with the expertise of a software system's evaluators. A process for the extraction of elements from source code and Analytical Hierarchical Processing for assigning weights to these data are provided; k-Attractors clustering algorithm is then applied on these data, in order to produce system overviews and deductions. The methodology is evaluated on Apache Geronimo, a large Open Source Application Server; results are discussed and conclusions are presented together with directions for future work.

Δημοσιεύσεις σε Περιοδικά & Συνέδρια

Διεθνή Επιστημονικά Περιοδικά

- D. ANTONIOU, J. GAROFALAKIS, C. MAKRIS, Y. PANAGIS AND E. SAKKOPOULOS, “CONTEXT-SIMILARITY BASED HOTLINKS ASSIGNMENT: MODEL, METRICS AND ALGORITHM.”, DATA AND KNOWLEDGE ENGINEERING (DKE), IN PRESS AVAILABLE ONLINE, (ISI SCIENCE CITATION INDEX (SCI)).
- D. ANTONIOU, E. SAKKOPOULOS, P. ADAMOPOULOU, N. TSIRAKIS AND A. TSAKALIDIS, “A WEB PERSONALIZING TECHNIQUE USING ADAPTIVE DATA STRUCTURES: THE CASE OF BURSTS IN WEB VISITS”, JOURNAL OF SYSTEMS AND SOFTWARE 2010 0164-1212.
- MARIANTHI GEORGITSI, EMMANOUIL VIENNAS, DIMITRIS ANTONIOU, VASSILIKI GKANTOUNA, SJOZEF VAN BAAL, EMANUEL F. PETRICOIN, KONSTANTINOS POULAS, GIANNIS TZIMAS, ATHANASIOS TSAKALIDIS AND GEORGE P. PATRINOS, “FINDBASE: A WORLDWIDE DATABASE FOR GENETIC VARIATION ALLELE FREQUENCIES UPDATED”, *NUCL. ACIDS RES.* (2010).
- ANTONELLIS P., ANTONIOU D., KANELLOPOULOS Y., MAKRIS C., THEODORIDIS E., TJORTJIS C., TSIRAKIS N., “CLUSTERING FOR MONITORING SOFTWARE SYSTEMS MAINTAINABILITY EVOLUTION”, ELECTRONIC NOTES IN THEORETICAL COMPUTER SCIENCE, (ELSEVIER), VOL. 233, MARCH 2009, PP. 43-57, ISSN 1571-0661.
- D. ANTONIOU, I. KAMPOURIS, E. THEODORIDIS. “SPLAYING FOR COMPRESSION: A THEORETICAL AND EXPERIMENTAL STUDY”, JOURNAL OF DISCRETE ALGORITHMS (JDA) (EXEI YΠΟΒΛΗΘΕΙ).
- D. ANTONIOU, E. SAKKOPOULOS, C. MAKRIS, “A WEB RECOMMENDATION ALGORITHM USING SPLAY TREES: THE CASE OF BURST OF VISITS”, DATA & KNOWLEDGE ENGINEERING, (DKE) (EXEI YΠΟΒΛΗΘΕΙ).
- KANELLOPOULOS Y., ANTONELLIS P., ANTONIOU D., MAKRIS C., THEODORIDIS E., TJORTJIS C., TSIRAKIS N., “CODE QUALITY EVALUATION METHODOLOGY USING THE ISO/IEC 9126 STANDARD”, INT’L JOURNAL OF SOFTWARE ENGINEERING & APPLICATIONS, VOL.1, NO.2, JULY 2010, PP. 17-36, (AIRCC), ISSN 0976-2221.

Διεθνή Επιστημονικά Συνέδρια

- DIMITRIS ANTONIOU, MERSINI PASCHOU, EFROSINI SOURLA, ATHANASIOS TSAKALIDIS, “A SEMANTIC WEB PERSONALIZING TECHNIQUE, THE CASE OF BURSTS IN WEB VISITS”, INTERNATIONAL WORKSHOP ON SEMANTIC WEB AND REASONING FOR CULTURAL HERITAGE AND DIGITAL LIBRARIES (SWARCH-DL), IN ICSC 2010.
- DIMITRIS ANTONIOU, EVANGELOS THEODORIDIS AND ATHANASIOS TSAKALIDIS, “COMPRESSING BIOLOGICAL SEQUENCES USING SELF ADJUSTING DATA STRUCTURES”, THE 10TH IEEE

- INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY AND APPLICATIONS IN BIOMEDICINE ,ITAB 2010.
- P. ANTONELLIS, D. ANTONIOU, Y. KANELLOPOULOS, C. MAKRIS, E. THEODORIDIS, C. TJORTJIS AND N. TSIRAKIS. "A DATA MINING METHODOLOGY FOR EVALUATING MAINTAINABILITY ACCORDING TO ISO/IEC-9126 SOFTWARE ENGINEERING-PRODUCT QUALITY STANDARD". SPECIAL SESSION ON SYSTEM QUALITY AND MAINTAINABILITY, ORGANIZED IN CONJUNCTION WITH THE 11TH EUROPEAN CONFERENCE ON SOFTWARE MAINTENANCE AND REENGINEERING, CSMR 2007.
- P. ANTONELLIS, D. ANTONIOU, Y. KANELLOPOULOS, C. MAKRIS, E. THEODORIDIS, C. TJORTJIS AND N. TSIRAKIS, "CLUSTERING FOR MONITORING SOFTWARE SYSTEMS MAINTAINABILITY EVOLUTION". EUROPEAN CONFERENCE ON SOFTWARE MAINTENANCE AND REENGINEERING (CSMR) 2008, INTERNATIONAL WORKSHOP ON SOFTWARE QUALITY AND MAINTAINABILITY (SQM 2008).
- PANOS ANTONELLIS, DIMITRIS ANTONIOU, YIANNIS KANELLOPOULOS, CHRISTOS MAKRIS, CHRISTOS TJORTJIS, VANGELIS THEODORIDIS, NIKOS TSIRAKIS, "CODE4THOUGHT PROJECT: EMPLOYING THE ISO/IEC-9126 STANDARD FOR SOFTWARE ENGINEERING-PRODUCT QUALITY ASSESSMENT," CSMR, pp.297-300, 2009 EUROPEAN CONFERENCE ON SOFTWARE MAINTENANCE AND REENGINEERING, 2009.
- ANTONELLIS P., ANTONIOU D., THEODORIDIS E., KANELLOPOULOS Y., MAKRIS C., TJORTJIS C., TSIRAKIS N., "EMPLOYING CLUSTERING FOR ASSISTING SOURCE CODE MAINTAINABILITY EVALUATION ACCORDING TO ISO/IEC-9126" ARTIFICIAL INTELLIGENCE TECHNIQUES IN SOFTWARE ENGINEERING WORKSHOP (AISEW 2008) IN ECAI08.

ΜΕΡΟΣ Α:

ΕΙΣΑΓΩΓΗ

Κεφάλαιο 1

Εισαγωγικές Έννοιες

1.1 Αλγόριθμοι και Δομές Δεδομένων

Δύσκολα θα μπορούσε να φανταστεί κάποιος μη ειδικός το ρόλο των αλγορίθμων και των δομών δεδομένων στην καθημερινή ζωή. Για τον περισσότερο κόσμο η επιστήμη των αριθμών ακόμα και τώρα δε έχει προφανείς πρακτικές εφαρμογές στην καθημερινότητα. Στην πραγματικότητα όμως, χρησιμοποιούνται καθημερινά εκατοντάδες εφαρμογές σύνθετων μαθηματικών μέσω της πληροφορικής: οι λογαριασμοί της πιστωτικής κάρτας, τα φανάρια που αλλάζουν από πράσινο σε κόκκινο και πολλά ακόμη λειτουργούν με σύνθετους αλγόριθμους. Οι αλγόριθμοι στην επιστήμη των υπολογιστών αποτελούν ουσιαστικά τη ραχοκοκαλιά των προγραμμάτων υπολογιστών, καθώς εκφράζουν μια ακολουθία βημάτων που εξετάζουν το υπό εξέταση γεγονός και καταλήγουν σε μια ενέργεια. Ένα παράδειγμα είναι ο τρόπος με τον οποίο παράγονται τα αποτελέσματα των πανελληνίων εξετάσεων για την τριτοβάθμια εκπαίδευση. Ο υπολογιστής που τα παράγει, απαντά μια σειρά από κατάλληλα ερωτήματα: τι βαθμούς έχει ο μαθητής, ποιες σχολές έχει επιλέξει και με ποια σειρά και ποιες είναι οι βάσεις των σχολών. Ο αλγόριθμος οδηγεί τον υπολογιστή βήμα βήμα στην ανάδειξη των τελικών αποτελεσμάτων.

Γενικότερα, οι αλγόριθμοι έχουν κατακλύσει τη ζωή του ανθρώπου. Όταν ένας φοιτητής κάνει αίτηση για κάποιο πρόγραμμα μεταπτυχιακών σπουδών, ένας αλγόριθμος υπολογίζει τα μόριά του με βάση τα προσόντα του, αλλά και τα προσόντα των υπολοίπων. Στο αεροδρόμιο, ένας άλλος αλγόριθμος καθορίζει τη σειρά απογείωσης του κάθε αεροπλάνου. Οι φωτεινοί σηματοδότες συντονίζονται από έναν αλγόριθμο, το ίδιο και οι υπολογιστές ταξιδιού των αυτοκινήτων. Και από τη στιγμή που οι υπολογιστές έχουν πλέον κυριαρχήσει στη ζωή μας, οι αλγόριθμοι γίνονται ολοένα και πιο σημαντικοί.

Στην εποχή του αχανούς κόσμου του Διαδικτύου, οι αλγόριθμοι δεν είναι απλώς σημαντικοί, αλλά καθοριστικοί στα πλαίσια οποιασδήποτε διαδικτυακής εφαρμογής. Πίσω από τα αποτελέσματα σε μια μηχανή αναζήτησης ή από τη φράση «οι χρήστες που αγόρασαν αυτό το προϊόν έχουν αγοράσει επίσης...» ενός ιστότοπου ηλεκτρονικού εμπορίου, κρύβεται ένας αλγόριθμος. Η ανάπτυξη και εφαρμογή αυτών των μαθηματικών μηχανισμών στο Διαδίκτυο, έχουν αποφέρει δυσθεώρητα κέρδη σε εταιρείες όπως η Google, και όπως ήταν αναμενόμενο οι ακριβείς λεπτομέρειές τους φυλάσσονται σαν επτασφράγιστα μυστικά, που χαρακτηρίζουν την εξέλιξη στον 21ο αιώνα. Αλλά όπου υπάρχουν κερδοφόρα μυστικά, υπάρχουν και ερευνητές που προσπαθούν απελπισμένα να τα ανακαλύψουν, βελτιώσουν και ξεπεράσουν. Σε ολόκληρο τον κόσμο, ξοδεύονται τεράστια ποσά και άπειρος χρόνος για να ξεκλειδωθούν οι πιο επιτυχημένοι αλγόριθμοι του Διαδικτύου.

Το μέλλον είναι απολύτως συνυφασμένο με τους αλγόριθμους και τις εφαρμογές τους, αν και η κατανόηση της λειτουργίας τους βρίσκεται ακόμη σε πολύ πρώιμο στάδιο, καθώς η λειτουργία τους είναι σχεδόν πάντα αόρατη και ακατανόητη προς τον πολύ κόσμο.

Με τη βοήθεια των αλγορίθμων, ο υπολογιστής ήδη αναγνωρίζει ακριβώς ποια πληροφορία ψάχνει ο χρήστης στο Διαδίκτυο και του προτείνει πράγματα που πιθανότατα να επιθυμεί να διαβάσει. Στο μέλλον, τα τεχνολογικά άλματα στην επιστήμη των υπολογιστών, θα επιτρέπουν στον υπολογιστή να μαντεύει τη διάθεσή μας και τις προθέσεις μας.

Η δομή δεδομένων είναι ένας τρόπος οργάνωσης στοιχείων σε έναν υπολογιστή έτσι ώστε αυτά να μπορούν να χρησιμοποιηθούν αποτελεσματικά και αποδοτικά. Συχνά μια προσεκτικά επιλεγμένη δομή δεδομένων οδηγεί στη υλοποίηση ενός αποδοτικότερου αλγορίθμου. Η επιλογή της δομής δεδομένων αρχίζει συχνά από την επιλογή ενός κατάλληλου τύπου στοιχείων. Μια καλά σχεδιασμένη δομή δεδομένων επιτρέπει σε ποικίλες κρίσιμες διαδικασίες να εκτελούνται χρησιμοποιώντας λιγότερους πόρους, οι οποίοι μεταφράζονται σε μικρότερο χρόνο εκτέλεσης και δέσμευση λιγότερου χώρου στη μνήμη.

Μέχρι σήμερα έχουν αναπτυχθεί διαφορετικά είδη δομών δεδομένων ώστε να ταιριάζουν στα διαφορετικά είδη εφαρμογών, και μερικές δομές είναι ιδιαίτερα εξειδικευμένες σε ορισμένους στόχους. Παραδείγματος χάριν, τα B-δέντρα βρίσκουν εφαρμογή στην ανάπτυξη βάσεων δεδομένων, ενώ τα R-trees χρησιμοποιούνται στα GIS συστήματα.

Στο σχεδιασμό πολλών τύπων προγραμμάτων, η επιλογή των δομών δεδομένων αποτελεί βασικό συστατικό στην αρχική εκτίμηση του σχεδίου, δεδομένου ότι η εμπειρία στην οικοδόμηση μεγάλων συστημάτων έχει δείξει ότι τόσο η δυσκολία στην υλοποίηση αλλά και η απόδοση και ποιότητα του τελικού αποτελέσματος εξαρτώνται σε μεγάλο βαθμό από την επιλογή της κατάλληλης δομής δεδομένων. Μετά την επιλογή των δομών δεδομένων, ο αλγόριθμος που θα χρησιμοποιηθεί επιλέγεται ευκολότερα και είναι συχνά σχετικά προφανής. Μερικές φορές τα πράγματα λειτουργούν στην αντίθετη κατεύθυνση - οι δομές δεδομένων επιλέγονται επειδή ορισμένοι βασικές διαδικασίες χρησιμοποιούν αλγορίθμους που λειτουργούν καλύτερα με συγκεκριμένες δομές δεδομένων. Σε κάθε περίπτωση, η επιλογή των κατάλληλων δομών δεδομένων είναι κρίσιμη.

Δεδομένου ότι οι δομές δεδομένων είναι τόσο χρήσιμες και καθοριστικές, πολλές από αυτές συμπεριλαμβάνονται σε τυποποιημένες βιβλιοθήκες σύγχρονων γλωσσών προγραμματισμού και περιβάλλοντα, όπως η Standard Template Library της C++, αλλά και τα Frameworks της JAVA και της Microsoft .NET.

Οι δομές δεδομένων αντιπροσωπεύουν εφαρμογές ή διεπαφές. Μια δομή δεδομένων μπορεί να αντιμετωπισθεί ως διεπαφή μεταξύ δύο λειτουργιών ή ως εφαρμογή των μεθόδων στην πρόσβαση αποθηκευμένων δεδομένων που είναι οργανωμένα σύμφωνα με το συναφή τύπο στοιχείων.

Στα πλαίσια αυτής της διδακτορικής διατριβής προτείνονται αποτελεσματικοί αλγόριθμοι και δομές δεδομένων με εφαρμογές στις τεχνολογίες διαδικτύου και στην ανάκτηση πληροφορίας. Συγκεκριμένα, στα κεφάλαια που ακολουθούν θα μελετηθούν δομές δεδομένων και αλγόριθμοι με σκοπό την εφαρμογή τους στη συμπίεση δεδομένων, την επανοργάνωση ιστότοπων, την εξατομίκευση του Διαδικτύου, την ανάπτυξη συστημάτων σύστασης ιστοσελίδων και στην αξιολόγηση λογισμικού.

1.2 Συμπύεση Δεδομένων

Στα πλαίσια αυτής της ενότητας, γίνεται μελέτη της εφαρμογής των Splay δέντρων στη συμπύεση δεδομένων. Η βασική δομή δεδομένων είναι τα δυαδικά δέντρα αναζήτησης. Η παρατήρηση των μειονεκτημάτων κάποιων αποδοτικών δένδρων αναζήτησης οδήγησε τους Tarjan και Sleator στον ορισμό του splay δένδρου [1]. Συγκεκριμένα, τα ισοζυγισμένα δένδρα, όπως είναι τα υψοζυγισμένα δένδρα (height-balanced trees) [2, 3], τα βαροζυγισμένα δένδρα (weight-balanced trees) [4], τα B-δένδρα (B-trees) [5] και οι παραλλαγές τους [6, 7, 8, 9], έχουν χρόνο χειρότερης περίπτωσης της τάξης του $O(\log n)$ ανά πράξη σε ένα δένδρο n κόμβων. Όμως τα ισοζυγισμένα δένδρα δεν είναι πλήρως αποδοτικά αν το μοτίβο πρόσβασης (access pattern) δεν είναι ομοιόμορφο, ενώ πρόσθετα χρειάζονται επιπλέον χώρο για την αποθήκευση της πληροφορίας ζύγισης (balance information). Τα βέλτιστα δυαδικά δένδρα [10, 11, 3] εγγυώνται ελάχιστο μέσο χρόνο προσπέλασης, αλλά μόνο υπό τον όρο ότι χρησιμοποιούνται σταθερές, γνωστές εκ των προτέρων, πιθανότητες προσπελάσεων και χωρίς καμία συσχέτιση ανάμεσα στις προσπελάσεις. Επιπλέον, το κόστος διαγραφής και εισαγωγής σε τέτοια δένδρα είναι πολύ υψηλό. Τα biased δένδρα αναζήτησης συνδυάζουν τον γρήγορο μέσο χρόνο προσπέλασης των βέλτιστων δένδρων με την ταχεία ανανέωση των ισοζυγισμένων δένδρων αλλά έχουν δομικά συστατικά ακόμα πιο πολύπλοκα και δύσκολο να πραγματοποιηθούν από τα ισοζυγισμένα δένδρα. Τα δυαδικά δένδρα δακτυλοδότησης (finger search trees) [8, 9, 12, 13, 14] επιτρέπουν γρήγορες προσπελάσεις στην περιοχή γύρω από το ένα ή τα περισσότερα “δάκτυλα” (fingers) αλλά απαιτούν την αποθήκευση επιπλέον δεικτών σε κάθε κόμβο.

Όλες αυτές οι δομές δεδομένων έχουν σχεδιαστεί με σκοπό τη μείωση του χρόνου χειρότερης περίπτωσης ανά πράξη. Όμως, σε τυπικές εφαρμογές των δυαδικών δένδρων εκτελείται ακολουθία πράξεων και όχι μία πράξη και αυτό που έχει σημασία είναι ο συνολικός χρόνος εκτέλεσης της ακολουθίας και όχι ο χρόνος κάθε πράξης ξεχωριστά. Σε τέτοιες εφαρμογές είναι προτιμότερη η ελάττωση των επιμερισμένων χρόνων των πράξεων, όπου με επιμερισμένο χρόνο εννοούμε τον μέσο χρόνο μιας πράξης σε μια χειρότερης περίπτωσης ακολουθία από πράξεις.

Ένας τρόπος για την επίτευξη αποδοτικότητας με την επιμερισμένη έννοια είναι η χρήση αυτοπροσαρμοζόμενης (self-adjusting) δομής δεδομένων. Η δομή βρίσκεται σε τυχαία αρχική κατάσταση αλλά κατά τη διάρκεια κάθε πράξης εφαρμόζεται ένας απλός ανακατασκευαστικός κανόνας με σκοπό την βελτίωση της αποδοτικότητας των μελλοντικών πράξεων. Παραδείγματα ανακατασκευαστικών κανόνων είναι ο move-to-front σε γραμμικές λίστες [17, 18] και η συμπύεση μονοπατιού (path compression) σε ισοζυγισμένα δένδρα [15, 16].

Οι αυτοπροσαρμοζόμενες δομές δεδομένων έχουν διάφορα πιθανά πλεονεκτήματα έναντι των ισοζυγισμένων ή αλλιώς των ρητά καταναγκασμένων δομών δεδομένων (explicitly constraint structures):

- i. Με την επιμερισμένη έννοια, αγνοώντας σταθερούς παράγοντες, δεν είναι ποτέ (οι αυτοπροσαρμοζόμενες δομές δεδομένων) πολύ χειρότερες από ότι οι καταναγκασμένες δομές, και εφόσον οι αυτοπροσαρμοζόμενες δομές προσαρμόζονται ανάλογα με τη χρήση, μπορούν να είναι πολύ πιο αποδοτικές αν το μοτίβο χρήσης είναι ανομοιόμορφο.

ii. Χρειάζονται λιγότερο χώρο αφού δεν αποθηκεύουν καμία πληροφορία ισορροπίας ή άλλη εξαναγκασμένη πληροφορία.

iii. Έχουν απλούς και εύκολα υλοποιήσιμους αλγόριθμους προσπέλασης και ανανέωσης.

Οι αυτοπροσαρμοζόμενες δομές δεδομένων έχουν δύο δυναμικά μειονεκτήματα:

i. Απαιτούν περισσότερες τοπικές αναπροσαρμογές, ειδικά κατά την διάρκεια των προσπελάσεων (πράξεων αναζήτησης). (Οι ρητά εξαναγκασμένες δομές χρειάζονται προσαρμογή μόνο κατά τη διάρκεια των ανανεώσεων, όχι κατά τη διάρκεια των προσπελάσεων).

ii. Οι ατομικές πράξεις μέσα σε μια ακολουθία μπορεί να είναι ακριβές, κάτι που μπορεί να είναι μειονέκτημα για εφαρμογές πραγματικού χρόνου (real-time applications).

Ο Douglas W. Jones [19] παρουσίασε μία εργασία πάνω στη συμπίεση δεδομένων που μπορεί να επιτευχθεί με τη χρήση splay δένδρων. Ο αλγόριθμος που σχεδίασε, καλείται αλγόριθμος splay-προθέματος (splay-prefix algorithm) και είναι ένας από τους γρηγορότερους και απλούστερους προσαρμοζόμενους αλγόριθμους συμπίεσης δεδομένων που βασίζεται στη χρήση προθεματικού κώδικα (prefix code). Οι δομές δεδομένων που χρησιμοποιούνται στον αλγόριθμο splay-προθέματος μπορούν επίσης να εφαρμοστούν στην αριθμητική συμπίεση δεδομένων. Οι αλγόριθμοι συμπίεσης δεδομένων μπορούν να βελτιώσουν την αποδοτικότητα με την οποία τα δεδομένα αποθηκεύονται ή μεταφέρονται, μέσω της μείωσης του ποσού της πλεονάζουσας πληροφορίας. Ένας αλγόριθμος συμπίεσης (compression algorithm) παίρνει σαν είσοδο ένα πηγαίο κείμενο και παράγει ένα αντίστοιχο συμπιεσμένο κείμενο, ενώ ένας αλγόριθμος αποσυμπίεσης (expansion algorithm) παίρνει το συμπιεσμένο κείμενο ως εισαγωγή και παράγει το αρχικό πηγαίο κείμενο σαν έξοδο. Τέτοιοι αλγόριθμοι καλούνται αλγόριθμοι χωρίς θόρυβο και με τέτοιους θα ασχοληθούμε σε αυτό το κεφάλαιο. Οι περισσότεροι αλγόριθμοι συμπίεσης βλέπουν το πηγαίο κείμενο να αποτελείται από μια ακολουθία χαρακτήρων (letters) που έχουν επιλεγεί από ένα αλφάβητο.

Στην παρούσα διατριβή, ο αλγόριθμος αυτός επαυξάνεται με τη χρήση μιας επέκτασης της splay τεχνικής, το chain splay, που προτάθηκε από τον George F. Georgakopoulos [20]. Τα δέντρα chain splay είναι $O(\log \log n)$ ανταγωνιστικά και επιτυγχάνουν επιμεριστικό χρόνο εκτέλεσης $O(\log n)$. Αυτό το αποτέλεσμα είναι το κοντινότερο στην εικασία Δυναμικής Βελτιστότητας των Splay δέντρων από το 1985.

1.3 Επανοργάνωση Δικτυακών Τύπων

Η ανάθεση «καυτών» υπερσυνδέσμων (hotlinks) σε ιστότοπους αποτελεί μία ακόμα προσπάθεια για βελτίωση της περιήγησης στο διαδίκτυο μέσω της καλύτερης σχεδίασης και οργάνωσης των ιστότοπων. Το σκεπτικό πίσω από αυτήν την προσπάθεια είναι να είναι ευκολότερα προσπελάσιμες από το χρήστη οι σελίδες που περιέχουν δημοφιλείς πληροφορίες και τις οποίες αναζητούν περισσότερο οι επισκέπτες του ιστότοπου. Ευκολότερη πρόσβαση σε μία ιστοσελίδα σημαίνει ότι είτε μείωση του μονοπατιού προς αυτήν την πληροφορία, δηλαδή λιγότερες μεταβάσεις, είτε μείωση του απαιτούμενου όγκου δεδομένων μέχρι να φτάσει ο χρήστης σε αυτήν την πληροφορία.

Με την ανάθεση hotlinks λοιπόν βελτιώνεται η πρόσβαση σε ιστοσελίδες του διαδικτύου βελτιώνοντας τη σχεδίαση των ιστότοπων. Ένας καλοσχεδιασμένος ιστότοπος αποφεύγει την περιττή μεταφορά δεδομένων κατά την αναζήτηση της επιθυμητής πληροφορίας, εξοικονομεί χρόνο για το χρήστη και μειώνει το φόρτο εργασίας συνολικά στο διαδίκτυο. Η ιδέα της ανάθεσης hotlinks λοιπόν, συνοψίζεται στο να έρθουν οι περισσότερο δημοφιλείς σελίδες πιο κοντά στην αρχική σελίδα του ιστότοπου, μειώνοντας έτσι το κόστος πρόσβασης σε αυτές.

Η ιδέα των hotlinks εισήχθει από τους Perkowski και Etzioni[21], όπου μεταξύ άλλων χρήσιμων μετασχηματισμών σε έναν ιστότοπο, πρότειναν το shortcutting που είναι η ανάθεση υπερσυνδέσμων από κάθε σελίδα προς τους τελικούς στόχους του επισκέπτη. Μετά από αυτή την αρχική εργασία, ακολούθησαν πολλές ακόμα εργασίες που μπορούν να κατηγοριοποιηθούν ανάλογα με το μοντέλο χρήστη. Υπάρχουν δύο μοντέλα που ακολουθούνται: Το μοντέλο του διορατικού χρήστη και το μοντέλο του άπληστου χρήστη. Στο πρώτο μοντέλο ο χρήστης έχει γνώση όλων των hotlinks που έχουν προστεθεί, δηλαδή γνωρίζει τη δομή ολόκληρου του ιστότοπου και είναι εφικτό γι' αυτόν να ακολουθήσει το συντομότερο μονοπάτι από την αρχική σελίδα προς την επιθυμητή πληροφορία. Στο δεύτερο μοντέλο, ο χρήστης έχει γνώση μόνο για τα hotlinks στην παρούσα ιστοσελίδα και εφόσον δεν έχει καθολική γνώση, παίρνει μια τοπική άπληστη απόφαση ακολουθώντας τον σύνδεσμο που τον οδηγεί στην επιθυμητή πληροφορία.

Γενικά, η μελέτη του προβλήματος της ανάθεσης hotlinks περιλαμβάνει ποικίλες παραμέτρους. Αρχικά, πρέπει να αποφασιστεί το όριο στον αριθμό hotlinks ανά σελίδα, η μετρική ανάδειξης της δημοτικότητας των ιστοσελίδων, ο τρόπος αναπαράστασης του ιστότοπου για την μαθηματική εξομοίωση του αλγορίθμου κ.ά.. Μία βασική παράμετρος στην ανάθεση hotlinks είναι ο καθορισμός του μέγιστου αριθμού hotlinks που μπορούν να ανατεθούν σε μία σελίδα. Με άλλα λόγια πόσα hotlinks μπορεί κάθε ιστοσελίδα να «δώσει» προς τις δημοφιλείς σελίδες. Με βάση αυτό το κριτήριο, οι αλγόριθμοι ανάθεσης hotlinks κατηγοριοποιούνται ως αλγόριθμοι ανάθεσης ενός hotlink ανά σελίδα ή k το πολύ hotlinks ανά σελίδα. Ένα άλλο σημαντικό κριτήριο ανάθεσης hotlinks, είναι η μετρική η οποία θα καθορίζει το ποιες σελίδες είναι δημοφιλείς και πόσο, ώστε ο εκάστοτε αλγόριθμος να βελτιώσει την πρόσβαση προς αυτές. Τέλος, όπως προαναφέρθηκε, ο τρόπος αναπαράστασης του ιστότοπου αποτελεί βασικό κριτήριο κατηγοριοποίησης των αλγορίθμων που έχουν προταθεί για την ανάθεση Hotlinks. Συγκεκριμένα, ένας ιστότοπος μπορεί να αναπαρασταθεί ως κατευθυνόμενος άκυκλος γράφος (DAG): $G = (V, E)$, όπου V το σύνολο των σελίδων του site που συνδέονται μεταξύ τους με ένα σύνολο E από links. Ένας άλλος τρόπος αναπαράστασης, ο οποίος είναι και ο πιο κοινός, είναι η αναπαράσταση ενός ιστότοπου ως ένα δυαδικό δέντρο, στο οποίο η ρίζα του δέντρου είναι η κεντρική ιστοσελίδα του ιστότοπου.

Ένας παράγοντας που είναι καθοριστικός και χαρακτηρίζει τους προτεινόμενους αλγόριθμους ανάθεσης hotlinks είναι η δυνατότητα της εφαρμογής τους σε πραγματικές συνθήκες. Οι περισσότερες εργασίες επικεντρώνονται σε μεγαλύτερο βαθμό σε θεωρητικό επίπεδο και κυρίως η απόδειξη άνω και κάτω ορίων στις πολυπλοκότητες χρόνου των προτεινόμενων αλγορίθμων. Απ' την άλλη, λίγες είναι οι εργασίες που ασχολούνται με την εφαρμογή των αλγορίθμων σε πραγματικές συνθήκες και πως θα μεταλλασσόταν η τοπολογία ενός ιστότοπου μετά την εφαρμογή. Για παράδειγμα, θα ήταν πρακτικά αδύνατο να αναπαρασταθεί ένας πραγματικός ιστότοπος με ένα δυαδικό δέντρο λόγω της διαφορετικής φύσης τους. Κάθε σελίδα περιέχει πολλούς συνδέσμους

προς άλλες σελίδες και όχι προς μόνο δύο όπως στην περίπτωση ενός δέντρου. Η ανάθεση hotlinks σε websites είναι ένα πρόβλημα με καθαρά πρακτικό χαρακτήρα, καθώς πηγάζει από τη δυσκολία της εύρεσης και ανάκτησης πληροφορίας από το διαδίκτυο, λόγω του μεγέθους του αλλά και της αναρχίας που χαρακτηρίζει την οργάνωσή του. Εν ολίγοις, η αποτελεσματικότητα του αλγορίθμου σε πραγματικές συνθήκες αλλά και η δυνατότητα της εφαρμογής του στην πράξη γενικότερα είναι το ίδιο και ίσως και παραπάνω σημαντικές από την χρονική πολυπλοκότητά του.

Στο κεφάλαιο 3 της παρούσας διδακτορικής διατριβής παρουσιάζεται ένας πρωτότυπος αλγόριθμος ανάθεσης Hotlinks. Πρόκειται για ένα τυχαιοποιημένο αλγόριθμο, ο οποίος συνδυάζει τη δημοτικότητα των ιστοσελίδων, τη δομή του δικτυακού τόπου, και για πρώτη φορά την ομοιότητα περιεχομένου μεταξύ των σελίδων, προκειμένου να προτείνει την ανάθεση των κατάλληλων υπερσυνδέσμων. Επιπλέον, βασίζεται στην χρησιμοποίηση πιο ρεαλιστικών γραφικών μοντέλων από το κλασικό μοντέλο δέντρου που χρησιμοποιείται μέχρι και τώρα. Με αυτόν τον τρόπο ο προτεινόμενος αλγόριθμος διαφοροποιείται από τις προηγούμενες εργασίες για την ανάθεση hotlinks.

1.4 Συστήματα Συστάσεων και Εξατομίκευσης Ιστότοπων

Τα τελευταία χρόνια γίνεται πολύς λόγος για τη χρήση συστημάτων συστάσεων (recommendation systems) στο διαδίκτυο. Ο σκοπός της ανάπτυξης συστημάτων συστάσεων είναι η προώθηση πληροφορίας στο χρήστη με κύριο γνώμονα τα ενδιαφέροντά του και τις προτιμήσεις του. Μέσα από τις επιλογές του χρήστη στο παρελθόν, το σύστημα αποκτάει γνώση και εμπειρία για την προσωπικότητά του και με βάση αυτή τη γνώση μπορεί να του κάνει συστάσεις που να ανταποκρίνονται σε αυτή.

Τα συστήματα αυτά αποτέλεσαν και αποτελούν ακόμη θέμα έρευνας και μελέτης καθώς βρίσκουν εφαρμογή σε πολλά πεδία, όπως οι εμπορικές δραστηριότητες, και τείνουν να μετατραπούν στις μέρες μας σε σημαντικά επιχειρησιακά εργαλεία. Με την ολοένα και αυξανόμενη συγκέντρωση δεδομένων στον παγκόσμιο ιστό και λαμβάνοντας υπόψιν τον τεράστιο όγκο πληροφορίας στον οποίο εκτίθεται ο χρήστης, τα συστήματα συστάσεων αποκτούν ακόμη μεγαλύτερη σημασία. Αρχικά, εμφανίστηκαν στα μέσα των '90s και χρησιμοποιούνταν μόνο από λίγα websites. Σήμερα πλέον δεν νοούνται δικτυακοί ιστότοποι e-εμπορίου που να μην ενσωματώνουν συστήματα συστάσεων με σκοπό να βοηθούν τους χρήστες τους στις αγορές τους. Με τη χρήση των συστημάτων αυτών, τα ηλεκτρονικά καταστήματα μπορούν να βελτιώσουν τις προσφερόμενες υπηρεσίες τους, ικανοποιώντας τις ανάγκες και τις προτιμήσεις κάθε πελάτη ξεχωριστά και διευκολύνοντας την πλοήγηση σε αυτά. Πρακτικά, αποτελούν ένα μέσο για να επιτύχει ο ιστότοπος να μετατρέψει τους επισκέπτες σε αγοραστές, να αυξάνει το cross-selling και το up-selling και να κερδίζει την εμπιστοσύνη και την πίστη των πελατών.

Γενικά, τα συστήματα συστάσεων μπορούν να κατηγοριοποιηθούν σε τρεις βασικές κατηγορίες:

1. φίλτράρισμα με βάση το περιεχόμενο,

2. συνεργατικό φιλτράρισμα και
3. υβριδικές προσεγγίσεις.

Το φιλτράρισμα με βάση το περιεχόμενο λαμβάνει υπόψιν τις πληροφορίες του χρήστη σε σχέση με τις πληροφορίες των προϊόντων (μεταδεδομένα, λέξεις κλειδιά, κλπ.) ενώ το συνεργατικό φιλτράρισμα την «ομοιότητα» των προφίλ των χρηστών προκειμένου να παράγουν συστάσεις. Οι δύο αυτές μέθοδοι μπορούν να συνδυαστούν μέσω υβριδικών προσεγγίσεων. Με άλλα λόγια, ένα σύστημα συστάσεων εκμεταλλεύεται τις προτιμήσεις των χρηστών αλλά και των κοινοτήτων χρηστών προκειμένου να βοηθήσει τον εκάστοτε χρήστη να ανακαλύψει τις πληροφορίες ή τα προϊόντα για τα οποία ενδιαφέρεται και είναι όσο το δυνατόν πιο κοντά στις ανάγκες και τις προτιμήσεις του. Οι συστάσεις μπορούν να λάβουν πολλές μορφές σε έναν ιστότοπο, όπως εξατομικευμένες προσφορές, προϊόντα, προσθήκη ή παράλειψη θεματικών ενοτήτων, αναδιάταξη, απόκρυψη, εμφάνιση, παράληψη ή προσθήκη συνδέσμων κλπ.

Τα συστήματα συστάσεων αποτελούν μέρος του συνόλου ενεργειών που έχουν σαν σκοπό την εξατομίκευση του διαδικτύου (web personalization). Σαν web personalization ορίζεται οποιαδήποτε ενέργεια που προσαρμόζει τις πληροφορίες ή τις υπηρεσίες που παρέχονται από ένα δικτυακό τόπο στις ανάγκες ενός συγκεκριμένου χρήστη ή ενός συνόλου χρηστών. Η διαδικασία της εξατομίκευσης περιλαμβάνει την προεπεξεργασία και κατηγοριοποίηση των δεδομένων, την εξαγωγή των συσχετίσεων μεταξύ τους και τον καθορισμό των ενεργειών που θα προταθούν.

Στην παρούσα διδακτορική διατριβή αντιμετωπίζεται το πρόβλημα όπου λίγες ιστοσελίδες γίνονται πολύ δημοφιλείς για σύντομα χρονικά διαστήματα και προσπελαύνονται πολύ συχνά σε ένα περιορισμένο χρονικό χώρο. Στόχος είναι αυτές οι εξάρσεις των επισκέψεων να διαχειριστούν και να προταθούν αυτές οι άκρως δημοφιλείς σελίδες στο μέλλον σε χρήστες που έχουν κοινά ενδιαφέροντα. Ως εκ τούτου, σε αυτό το κεφάλαιο, προτείνεται μια νέα τεχνική εξατομίκευσης με βάση προηγμένες δομές δεδομένων. Οι δομές δεδομένων που χρησιμοποιούνται είναι το Splay Δέντρο (1) και οι δυαδικοί σωροί (2). Περιγράφονται η αρχιτεκτονική της τεχνικής, η ανάλυση του χρόνου και η πολυπλοκότητα του χώρου και αποδεικνύεται η απόδοσή της. Επίσης, συγκρίνονται θεωρητικά και πειραματικά η προτεινόμενη τεχνική με μια άλλη προσέγγιση για την επαλήθευση της αποτελεσματικότητάς της. Η προτεινόμενη λύση επιτυγχάνει $O(P^2)$ πολυπλοκότητα χώρου και τρέχει σε $k \cdot \log P$ χρόνο, όπου k είναι ο αριθμός των σελίδων και P ο αριθμός των κατηγοριών των ιστοσελίδων.

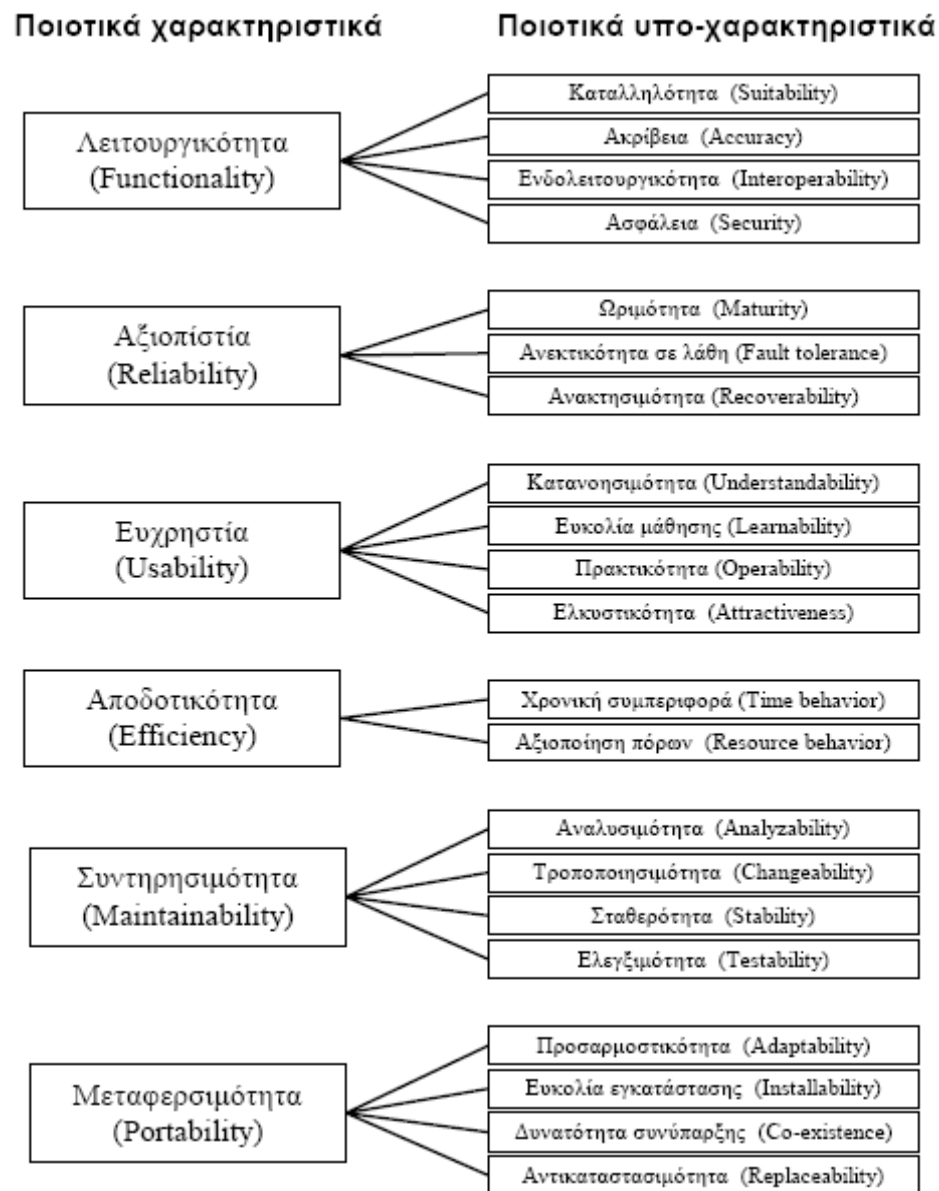
Επιπλέον, προτείνεται ένας αποτελεσματικός αλγόριθμος ανεύρεσης εξάρσεων επισκέψεων σε ιστοσελίδες με σκοπό την αξιοποίηση της πληροφορίας σε συστήματα σύστασης αλλά και σε άλλες εφαρμογές. Συγκεκριμένα, προτείνεται ένας αλγόριθμος για την εξαγωγή των "bursty" ιστοσελίδων ενός ιστότοπου με βάση τον αριθμό των επισκέψεων της κάθε ιστοσελίδας σε ένα ορισμένο χρονικό διάστημα. Με βάση αυτή τη λογική, παρέχεται η δυνατότητα να επιλεχθεί το χρονικό διάστημα που μας απασχολεί αλλά και το μέγεθος του burst, δηλαδή ο αριθμός των επισκέψεων αλλά και το χρονικό διάστημα που υποδηλώνουν ένα burst. Η δομή των δεδομένων που χρησιμοποιείται για τους σκοπούς του αλγόριθμου είναι και σε αυτή την περίπτωση το Splay δέντρο.

1.5 Αξιολόγηση ποιότητας πηγαίου κώδικα

Τέλος, όσον αφορά το πεδίο της ανάκτησης πληροφορίας, η διατριβή επικεντρώνεται σε μία πρωτότυπη και ολοκληρωμένη μεθοδολογία με σκοπό την αξιολόγηση της ποιότητας ενός συστήματος λογισμικού βάσει του Προτύπου Ποιότητας ISO/IEC-9126.

1.5.1 Το πρότυπο ISO9126

Το πρότυπο αυτό [22] είναι το αποτέλεσμα της διεθνούς προσπάθειας να αναπτυχθεί ένα πρότυπο για τη μέτρηση της ποιότητας λογισμικού. Είχε ως βάση το μοντέλο του McCall και αρχικά ονομάστηκε “Software Product Evaluation: Quality Characteristics and Guidelines for their Use”, το οποίο αργότερα εξελίχθηκε στο πρότυπο ISO9126. Σύμφωνα με αυτό, η ποιότητα καθορίζεται ως το σύνολο των δομικών στοιχείων και των χαρακτηριστικών ενός προϊόντος λογισμικού που επηρεάζουν την ικανότητά του να ικανοποιεί συγκεκριμένες απαιτήσεις και ανάγκες [23]. Αρχικά η ποιότητα χωρίζεται σε έξι ποιοτικά χαρακτηριστικά που υποδιαιρούνται σε ένα σύνολο από υπο-χαρακτηριστικά. Η περαιτέρω διάσπαση των ποιοτικών χαρακτηριστικών σε υποχαρακτηριστικά προκειμένου να μετρηθεί η ποιότητα [24] είναι μεγάλης σημασίας. Η διάσπαση της ποιότητας σε χαρακτηριστικά και υπο-χαρακτηριστικά σύμφωνα με το πρότυπο ISO9126 παρουσιάζεται στο Σχήμα.



Εικόνα 1 Η διάσπαση της ποιότητας σε χαρακτηριστικά και υπο-χαρακτηριστικά σύμφωνα με το πρότυπο ISO9126

Οι μετρήσεις στο λογισμικό γίνονται με τη χρήση μετρικών. Η χρήση μετρικών στην τεχνολογία λογισμικού αντιμετωπίζει την αδυναμία του λογισμικού για σαφή καθορισμό μετρήσιμων ποσοτήτων κατά τη σχεδίαση και ανάπτυξη ενός έργου. Πολλές φορές διάφορες εταιρείες παραγωγής λογισμικού υπόσχονται ότι τα προγράμματά τους είναι

π.χ. αξιόπιστα ή φιλικά προς το χρήστη, χωρίς όμως να έχουν προσδιορίσει εκ των προτέρων τι σημαίνει καθένα από αυτά με μετρήσιμες ποσότητες.

Μία πρώτη βασική κατηγοριοποίηση [25] χωρίζει τις μετρήσεις και συνεπώς και τις μετρικές λογισμικού σε:

1. Μετρικές Διαδικασιών, που αφορούν κάθε δραστηριότητα που σχετίζεται με την παραγωγή λογισμικού. Συνήθως εμπεριέχουν τον παράγοντα του χρόνου. Παραδείγματα στα οποία εφαρμόζονται: χρόνος εκτέλεσης μίας διαδικασίας, προσπάθεια, αριθμός εμφάνισης γεγονότων ενός συγκεκριμένου τύπου, δυνατότητα ελέγχου κ.ά.

2. Μετρικές Προϊόντων, που αφορούν κάθε παραγόμενο, παραδοτέο ή κείμενο που προκύπτει από μία διαδικασία. Παραδείγματα στα οποία εφαρμόζονται: μήκος προγράμματος, τμηματοποίηση, λειτουργικότητα, συνοχή και σύνδεση τμημάτων λογισμικού, εγχειρίδια χρήσης, υπηρεσίες κ.ά.

3. Μετρικές Πόρων, που αφορούν οντότητες που αποτελούν δεδομένα εισόδου στις διαδικασίες. Παραδείγματα στα οποία εφαρμόζονται: μέγεθος ομάδων εργασίας, εμπειρία, ηλικία, κόστος, δομή, επικοινωνία, υλικό κ.ά.

Μία άλλη κατηγοριοποίηση των μετρικών λογισμικού γίνεται με βάση τα χαρακτηριστικά του λογισμικού στα οποία αναφέρονται: τα εσωτερικά και τα εξωτερικά. Εσωτερικά χαρακτηριστικά του λογισμικού είναι τα χαρακτηριστικά για τα οποία υπάρχει απτή φυσική αντίληψη και άμεσος τρόπος μέτρησής τους (π.χ. γραμμές κώδικα, ποσοστό σχολίων στον κώδικα, αριθμός εντολών διακλάδωσης κ.ά.). Αυτά τα χαρακτηριστικά μετρούνται εύκολα και αυτόματα, αλλά δεν προσφέρουν πληροφορίες υψηλού επιπέδου που σχετίζονται με την ποιότητα του προϊόντος. Εξωτερικά χαρακτηριστικά του λογισμικού είναι τα χαρακτηριστικά με υψηλό επίπεδο αφαιρετικότητας τα οποία συνθέτουν την ποιότητα του λογισμικού (π.χ. λειτουργικότητα, αξιοπιστία, αποδοτικότητα, συντηρησιμότητα κ.ά.). Παρόλο που αυτά τα χαρακτηριστικά δίνουν ακριβείς πληροφορίες για την ποιότητα του λογισμικού, δυστυχώς είναι δύσκολο έως αδύνατον να μετρηθούν άμεσα.

Αντίστοιχα με τα ποιοτικά χαρακτηριστικά λοιπόν ορίζονται και οι μετρικές λογισμικού. Συγκεκριμένα έχουμε τις:

1. Εσωτερικές μετρικές, που είναι οι μετρικές αυτές που χρησιμοποιούνται για τη μέτρηση χαρακτηριστικών του λογισμικού για τα οποία υπάρχει απτή αντίληψη για τη φυσική τους σημασία και δυνατότητα άμεσης μέτρησης.

2. Εξωτερικές μετρικές, που είναι οι μετρικές αυτές που χρησιμοποιούνται για τη μέτρηση χαρακτηριστικών του λογισμικού με υψηλό επίπεδο αφαίρεσης τα οποία σχετίζονται άμεσα με την ποιότητα του λογισμικού.

Οι εσωτερικές μετρικές μπορούν να υπολογιστούν εύκολα και αυτόματα, αλλά το ζητούμενο δεν είναι τόσο ο τρόπος μέτρησής τους, όσο η σύνδεσή τους με τα εξωτερικά χαρακτηριστικά του λογισμικού τα οποία τελικώς επιθυμούμε να μετρήσουμε. Οι εξωτερικές μετρικές βασίζονται αποκλειστικά σε έρευνες γνώμης των πελατών-χρηστών με τη βοήθεια συνήθως ερωτηματολογίων και συνεντεύξεων. Αν και σύμφωνα με τον ορισμό της μετρικής που προηγήθηκε, οι εξωτερικές μετρικές δεν είναι κανονικές μετρικές, αρκετοί ερευνητές τις αναφέρουν και τις χρησιμοποιούν ως μετρικές ποιότητας.

1.5.2 Εκτίμηση της συντηρησιμότητας λογισμικού

Σε αυτή τη διατριβή θα δοθεί βάρος στην εφαρμογή μεθόδων εξόρυξης δεδομένων πάνω στα αποτελέσματα της μέτρησης μετρικών οι οποίες συνθέτουν τα χαρακτηριστικά του πηγαιού κώδικα, όπως αυτά ορίζονται από το Προτότυπο Ποιότητας ISO/IEC-9126. Ειδικότερα εφαρμόζονται αλγόριθμοι συσταδοποίησης με σκοπό την εύρεση τμημάτων κώδικα με ιδιαίτερα χαρακτηριστικά, που χρήζουν προσοχής. Συγκεκριμένα, παρουσιάζεται μια μεθοδολογία που βοηθά στην εκτίμηση της συντηρησιμότητας ενός προϊόντος λογισμικού σύμφωνα με το πρότυπο ποιότητας λογισμικού ISO/IEC-9126. Το κίνητρο πίσω από αυτή τη μεθοδολογία είναι η ολοκλήρωση των ομάδων των δεδομένων που εξάγονται από τη μέτρηση μετρικών που αφορούν τον πηγαιό κώδικα, με την εμπειρογνωμοσύνη των εκτιμητών του συστήματος λογισμικού, παρέχοντάς τους τη δυνατότητα να ορίσουν ένα σύνολο χαρακτηριστικών, κατάλληλων για μια τέτοιου είδους εκτίμηση.

Βιβλιογραφία Κεφαλαίου

1. D.D. Sleator and R.E. Tarjan. Self-adjusting binary search trees, Journal of the ACM, 32:652-686, 1985.
2. G.M. Adel'son – Vel'skii and E.M. Landis. An algorithm for the organization of information. Soviet Math, Dokl., 3:1259-1262, 1962.
3. D. E. Knuth. The Art of Computer Programming, Sorting and Searching, volume 3. Addison – Wesley
4. J. Nievergelt and E. M. Reingold. Binary search trees of bounded balance. SIAM J.Comput.2, pages 33-43, 1973.
5. R. Bayer and E McCreight. Organization of large ordered indexes. Acta Inf. 1 pages 173-189, 1972.
6. R. Bayer. Symmetric binary B-trees: Data structure and maintenance algorithms. Acta Inf. 1 pages 290-306, 1972.
7. S. Huddleston and K. Mehlhorn. Robust balancing in B-trees. In Theoretical Computer Science: 5th GI-Conference (Karlseuhe, West Germany, March 23-25). Lecture Notes in Computer Science, vol. 104. Springer-Verlag, Berlin, 1981, pages 234-244.
8. S. Huddleston and K. Mehlhorn. A new data structure for representing sorted lists. Acta Inf. 17, pages 157-184, 1982.
9. D. Majer and S. Salveter. Hysterical B-trees. Inf. Proc. Lett. 12, pages 199-202, 1981.
10. T. C. Hu and A. C. Tucker. Optimal computer-search trees and variable length alphabetic codes. SIAM J. App. Math. 37, pages 246-256, 1979.
11. D.E. Knuth. Optimum binary search trees. Acta Informatica, pages 14-25, 1971.
12. M. R. Brown and R. E. Tarjan. Design and analysis of a data structure for representing sorted lists. SIAM J. Comput 9, pages 594-614, 1980.
13. L. G. Guibas, E. M. Plass and J. R. Roberts. A new representation for linear lists. In Proceedings of the 9th Symposium on Theory of Computing. ACM, New York, pages 49-60, 1977.
14. S.R. Kosaraju. Localized search in sorted lists. In Proceedings of the 13th Annual ACM Symposium on Theory of Computing. ACM, New York, pages 62-69, 1981.

15. R. E. Tarjan. Efficiency of a good but not linear set union algorithm. J. ACM 22, pages 215-225, 2 1975.
16. R. E. Tarjan and J. Van Leeuwen. Worst-case analysis of set union algorithms. J. ACM 31, pages 245-281, 2 Apr.1984.
17. D.D.Sleator and R.E.Tarjan. Amortized efficiency of list update and paging rules. Communication of the ACM, 28Q202-208, 1985.
18. J.L. Bentley and C.C. McGeoch. Amortized analyses of self-organizing sequential search heuristics. Communication of the ACM, 28:404-411, 1985.
19. Douglas W. Jones: Application of Splay Trees to Data Compression. Commun. ACM 31(8): 996-1007 (1988)
20. George F. Georgakopoulos: How to Splay for loglog N-Competitiveness. WEA 2005: 570-579
21. Mike Perkowitz and Oren Etzioni. Adaptive Sites: an AI challenge. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence IJCAI 97, volume 1, pages 16-23, Nagoya, Japan, August 23-29 1997.
22. ISO/IEC 9126, “Software Product Evaluation – Quality Characteristics and Guidelines for the User”, International Organization for Standardization, Geneva, 2001.
23. Fenton N. and Pfleeger S., “Software Metrics”, PWS International Company, Thomson Computer Press, 1994.
24. Bache R. and Bazzana G., “Software Metrics for Product Assesment”, McGraw-Hill, 1994.
25. Fenton N.E., “*Software Metrics. A Rigorous Approach*”,Chapman & Hall, ISBN: 0-442-31355-1, 1992.
26. D.Antoniou , I. Kampouris, E.Theodoridis . “Splaying for Compression: A theoretical and Experimental Study”, Journal of Discrete Algorithms (JDA) (έχει υποβληθεί).
27. Dimitris Antoniou, Evangelos Theodoridis and Athanasios Tsakalidis, “Compressing Biological Sequences using Self Adjusting Data Structures”, The 10th IEEE International Conference on Information Technology and Applications in Biomedicine ,(ITAB 2010).



ΑΠΟΤΕΛΕΣΜΑΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΑΝΑΚΤΗΣΗ
ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΟ ΔΙΑΔΙΚΤΥΟ

Μέρος Β: Χρήση Αυτοοργανώμενων Δομών Δεδομένων στη Συμπύεση Δεδομένων

Κεφάλαιο 2

Εφαρμογή των Splay Δέντρων στη Συμπύεση Δεδομένων

Σύνοψη

Μια αυτοοργανώμενη δομή δεδομένων διαθέτει κάποιον αλγόριθμο για να αναδιοργανώνει τους δείκτες και τα δεδομένα κατάστασης μετά από κάθε πρόσβαση ή πράξη. Ο αλγόριθμος αυτοοργάνωσης είναι σχεδιασμένος ώστε αντιδρώντας σε αρχικά άγνωστες ιδιότητες της ακολουθίας αιτήσεων (request sequence), να οδηγεί τη δομή δεδομένων σε κατάσταση πλεονεκτική για τις ιδιότητες της ακολουθίας με αποτέλεσμα τη μείωση του χρόνου που χρειάζεται στο μέλλον ανά πράξη.

Ο πρώτος αλλά και ο μόνος μέχρι σήμερα πιθανός υποψήφιος αλγόριθμος αναζήτησης σε δένδρο που μπορεί να είναι $O(1)$ -ανταγωνιστικός είναι το splay δένδρο (splay tree) που παρουσιάστηκε από τους Sleator και Tarjan [1]. Ο George F. Georgakopoulos παρουσιάζει μια επέκταση της splay τεχνικής, την οποία ονομάζει chain-splay (αλυσιδωτό splay) [2]. Τα chain-splay δέντρα εφαρμόζουν splay στο στοιχείο που προσπελαύνουμε προς τη ρίζα όπως ακριβώς γίνεται και στα κλασικά splay trees, αλλά εκτελούν και μερικές τοπικές splay πράξεις τακτοποίησης κάτω από το στοιχείο που προσπελάστηκε. Αποδεικνύεται πως η τεχνική chain-splay είναι $O(\log \log n)$ ανταγωνιστική σε σχέση με οποιοδήποτε offline αλγόριθμο αναζήτησης. Τέλος, ο George F. Georgakopoulos [2] έδωσε ένα νέο λήμμα επαναζύγισης για τα splay δέντρα και με βάση αυτό το λήμμα, αποδεικνύει πως τα splay δέντρα είναι ανταγωνιστικά προς κάθε κλάση δυναμικών ισοζυγισμένων δέντρων.

Στη συνέχεια, γίνεται η παρουσίαση της δουλειάς του Douglas W. Jones [3] πάνω στη συμπίεση δεδομένων που μπορεί να επιτευχθεί με τη χρήση splay δέντρων. Ο αλγόριθμος που σχεδίασε, καλείται αλγόριθμος splay-προθέματος (splay-prefix algorithm) και είναι ένας από τους γρηγορότερους και απλούστερους προσαρμοζόμενους αλγόριθμους συμπίεσης δεδομένων που βασίζεται στη χρήση προθεματικού κώδικα (prefix code). Οι δομές δεδομένων που χρησιμοποιούνται στον αλγόριθμο splay-προθέματος μπορούν επίσης να εφαρμοστούν στην αριθμητική συμπίεση δεδομένων. Η χρήση αυτών των αλγορίθμων είναι αποδοτική τόσο στην κρυπτογράφηση όσο και στην επεξεργασία εικόνας.

Με σκοπό τη βελτίωση της απόδοσης του splay-prefix αλγορίθμου συμπίεσης, γίνεται αντικατάσταση της κλασικής splay δομής με την τεχνική chain-splay. Εφαρμόζεται συμπίεση σε βιολογικά δεδομένα αλλά και σε κείμενα τόσο με χρήση του κλασσικού splay δέντρου αλλά και της $\log \log n$ ανταγωνιστικής παραλλαγής του. Επιπλέον, παρουσιάζονται τυχαιοποιημένες εκδόσεις των παραπάνω δομών και εφαρμόζονται και αυτές στη συμπίεση δεδομένων. Οι $\log \log n$ ανταγωνιστικές δομές έχουν καλύτερη απόδοση όσον αφορά την πολυπλοκότητά τους σε σχέση με την κλασσική splay δομή και αναμένεται να επιτυγχάνουν καλύτερη συμπίεση. Το γεγονός αυτό επιβεβαιώνεται πειραματικά, όπου η επιτυγχανόμενη συμπίεση είναι στις περισσότερες των περιπτώσεων καλύτερη από την αντίστοιχη της κλασσικής δομής.

2.1 Splay δέντρα

Το splay δένδρο (splay tree) των Daniel Dominic Sleator και Robert Endre Tarjan [1], είναι μια αυτοοργανώμενη μορφή των δυαδικών δένδρων αναζήτησης. Τα δυαδικά δένδρα αναζήτησης είναι μια δομή δεδομένων για την αναπαράσταση πινάκων και λιστών με τέτοιο τρόπο, ώστε να είναι εύκολη η εισαγωγή, η προσπέλαση και η διαγραφή αντικειμένων. Σε ένα splay δέντρο n κόμβων, όλες οι κλασικές λειτουργίες ενός δένδρου αναζήτησης έχουν επιμερισμένο κόστος τάξης $O(\log n)$ ανά λειτουργία. Για το λόγο αυτό, τα δένδρα αυτά είναι τόσο αποδοτικά όσο τα ισοζυγισμένα δένδρα (balanced trees), όταν κύριο ρόλο παίζει ο συνολικός χρόνος εκτέλεσης. Επιπρόσθετα, για αποδοτικά μεγάλες ακολουθίες προσπελάσεων, τα splay δένδρα είναι τόσο αποδοτικά, κατά ένα σταθερό παράγοντα, όσο τα βέλτιστα στατικά ισοζυγισμένα δένδρα. Η αποδοτικότητα των splay δένδρων δεν οφείλεται σε ένα ρητό δομικό καταναγκασμό, όπως των ισοζυγισμένων δένδρων, αλλά από την εφαρμογή ενός απλού ευριστικού ανακατασκευής, το οποίο καλείται splaying, οποτεδήποτε υπάρχει προσπέλαση στο δένδρο.

Οι αυτοπροσαρμοζόμενες δομές δεδομένων έχουν διάφορα πιθανά πλεονεκτήματα έναντι των ισοζυγισμένων ή αλλιώς των ρητά καταναγκασμένων δομών δεδομένων (explicitly constraint structures):

1. Με την επιμερισμένη έννοια, αγνοώντας σταθερούς παράγοντες, δεν είναι ποτέ (οι αυτοπροσαρμοζόμενες δομές δεδομένων) πολύ χειρότερες από ό,τι οι καταναγκασμένες δομές, και εφόσον οι αυτοπροσαρμοζόμενες δομές προσαρμόζονται ανάλογα με τη χρήση, μπορούν να είναι πολύ πιο αποδοτικές αν το μοτίβο χρήσης είναι παράξενο.
2. Χρειάζονται λιγότερο χώρο αφού δεν αποθηκεύουν καμία πληροφορία ισορροπίας ή άλλη πληροφορία.
3. Έχουν απλούς και εύκολα υλοποιήσιμους αλγόριθμους προσπέλασης και ανανέωσης.

Οι αυτοπροσαρμοζόμενες δομές δεδομένων έχουν δύο δυναμικά μειονεκτήματα:

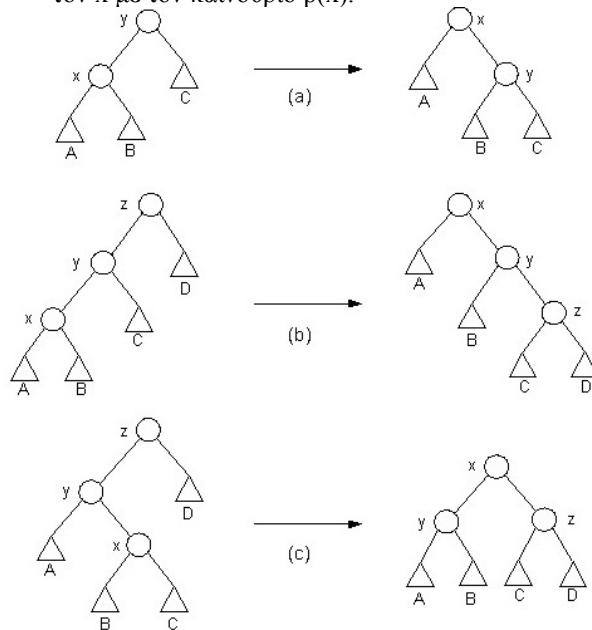
1. Απαιτούν περισσότερες τοπικές αναπροσαρμογές, ειδικά κατά την διάρκεια των προσπελάσεων (πράξεων αναζήτησης). (Οι ρητά εξαναγκασμένες δομές χρειάζονται προσαρμογή μόνο κατά τη διάρκεια των ανανεώσεων, όχι κατά τη διάρκεια των προσπελάσεων).
2. Οι ατομικές πράξεις μέσα σε μια ακολουθία μπορεί να είναι ακριβές, κάτι που μπορεί να είναι μειονέκτημα για εφαρμογές πραγματικού χρόνου (real-time applications).

Έστω ότι θέλουμε να φέρουμε εις πέρας μια ακολουθία προσπελάσεων σε ένα δυαδικό δένδρο αναζήτησης. Για να είναι μικρός ο συνολικός χρόνος προσπέλασης, τα αντικείμενα που ζητούνται συχνά, θα πρέπει να είναι συχνά κοντά στη ρίζα του δένδρου. Σκοπός είναι η κατασκευή ενός απλού τρόπου αναδόμησης του δένδρου, όπου μετά από κάθε προσπέλαση, να μεταφέρει το προσπελασμένο αντικείμενο πιο κοντά στη ρίζα, με την αληθοφανή υπόθεση ότι αυτό το αντικείμενο είναι πιθανό να ζητηθεί ξανά σύντομα.

Το αναδρομικό ευρετικό των Sleator και Tarjan καλείται *splaying* και είναι παρόμοιο με το *move-to-root* στο ότι κάνει περιστροφές από κάτω προς τα πάνω (bottom-up) κατά μήκος του μονοπατιού προσπέλασης (access path) και μετακινεί το προσπελασμένο αντικείμενο μέχρι τη ρίζα. Διαφέρει όμως, στο ότι κάνει περιστροφές κατά ζευγάρια, με σειρά που εξαρτάται από τη δομή του μονοπατιού προσπέλασης. Για να κάνουμε *splay* σε ένα δένδρο στον κόμβο x , επαναλαμβάνουμε το ακόλουθο βήμα *splaying* μέχρι το x να γίνει ρίζα του δένδρου (βλέπε Εικόνα 3).

Βήμα *Splaying*

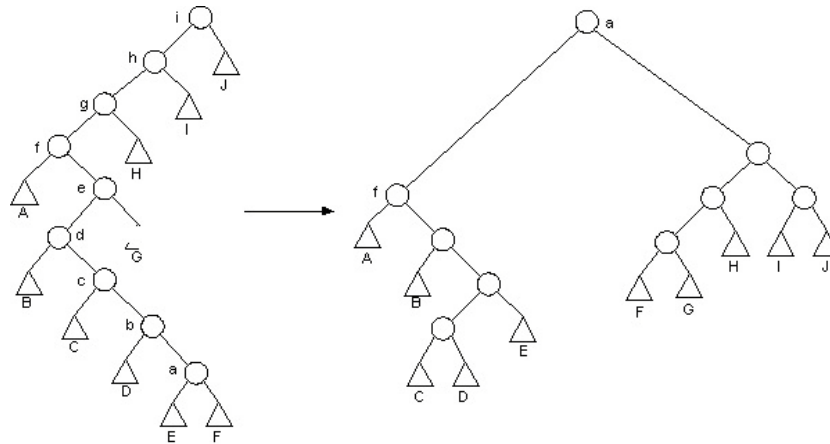
- Περίπτωση 1 (zig). Αν ο $p(x)$ (ο πατέρας του x) είναι ρίζα του δένδρου, κάνε περιστροφή της ακμής που ενώνει τον x με τον $p(x)$. (Αυτή η περίπτωση είναι τερματική.)
- Περίπτωση 2 (zig-zig). Αν ο $p(x)$ δεν είναι η ρίζα και οι x και $p(x)$ είναι και οι δύο αριστερά (ή και οι δύο δεξιά) παιδιά, κάνε περιστροφή της ακμής που ενώνει τον $p(x)$ με τον $g(x)$ (παππού του x) και στη συνέχεια κάνε περιστροφή της ακμής που ενώνει τον x με τον $p(x)$.
- Περίπτωση 3 (zig-zag): Αν ο $p(x)$ δεν είναι η ρίζα και ο x είναι δεξί παιδί και ο $p(x)$ αριστερό (ή αντίστροφα), κάνε περιστροφή της ακμής που ενώνει το x με τον $p(x)$ και στη συνέχεια κάνε περιστροφή της ακμής που ενώνει τον x με τον καινούριο $p(x)$.



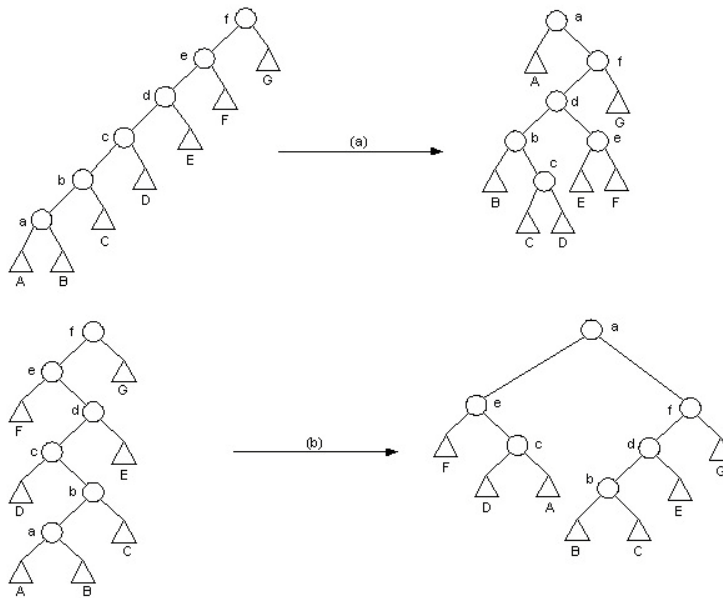
Εικόνα 2 Το βήμα Splaying. Ο κόμβος που προσπελάσσεται είναι ο x . (a) Zig, (b) zig-zig, (c) zig-zag.

Η *splaying* διαδικασία σε έναν κόμβο x βάθους d απαιτεί $\Theta(d)$ χρόνο, ο οποίος χρόνος είναι ανάλογος του χρόνου που χρειάζεται για να γίνει προσπέλαση στο αντικείμενο του x . Το *splaying* δε μεταφέρει απλά το x στη ρίζα, αλλά μπορούμε με

πρόχειρο τρόπο να πούμε ότι μειώνει στο μισό το βάθος κάθε κόμβου στο μονοπάτι προσπέλασης (βλέπε Εικόνες 3 και 4). Αυτή η ιδιότητα του splaying το κάνει αποδοτικό και δεν τη συναντάμε σε άλλα απλούστερα ευρετικά, όπως είναι το move-to-root. Είναι αυτή η ιδιότητα που απαιτεί (για να παραχθεί) το διαφορετικό χειρισμό των zig-zig και zig-zag περιπτώσεων.



Εικόνα 3 Splaying στον κόμβο α.

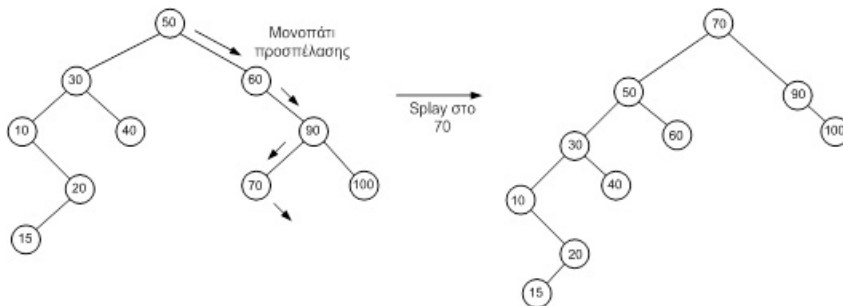


Εικόνα 4 Ειδικές περιπτώσεις Splaying. (a) Zig-zig μόνο και η τερματική. (b) Zig-zag μόνο και η τερματική.

Μπορούμε να υλοποιήσουμε όλες τις βασικές πράξεις ανανέωσης σε ένα δυαδικό δένδρο αναζήτησης χρησιμοποιώντας το splaying.

- $\text{access}(i, t)$: Αν το αντικείμενο i είναι στο δένδρο t , επέστρεψε ένα δείκτη στη θέση του. Αν όχι επέστρεψε ένα δείκτη στον κενό (null) κόμβο.
- $\text{insert}(i, t)$: Εισήγαγε το αντικείμενο i στο δένδρο t , υποθέτοντας ότι δεν είναι ήδη στο δένδρο.
- $\text{delete}(i, t)$: Διέγραψε το αντικείμενο i από το δένδρο t , υποθέτοντας ότι υπάρχει.
- $\text{join}(t_1, t_2)$: Συνδύασε τα δένδρα t_1 και t_2 σε ένα μόνο δένδρο που περιέχει όλα τα αντικείμενα και από τα δύο δένδρα και επέστρεψε το δένδρο που έχεις σαν αποτέλεσμα. Η πράξη υποθέτει ότι όλα τα αντικείμενα στο t_1 είναι μικρότερα από όλα τα αντικείμενα στο t_2 και καταστρέφει τα δένδρα t_1 και t_2 .
- $\text{split}(i, t)$: Κατασκεύασε και επέστρεψε δύο δένδρα t_1 και t_2 , όπου το t_1 να περιέχει όλα τα αντικείμενα στο t που είναι μικρότερα ή ίσα με το i , και το t_2 να περιέχει όλα τα αντικείμενα του t που είναι μεγαλύτερα από i . Αυτή η πράξη καταστρέφει το δένδρο t .

Αυτές οι πράξεις μπορούν να γίνουν και με splay δένδρα. Για να γίνει η $\text{access}(i, t)$, αρχικά αναζητούμε από τη ρίζα του δένδρου t και προς τα κάτω τον κόμβο i . Αν η αναζήτηση φτάσει σε κόμβο x που περιέχει το i , τερματίζουμε την access κάνοντας splaying στον κόμβο x και επιστρέφοντας έναν δείκτη στον x . Αν η αναζήτηση καταλήξει στον κενό κόμβο, το οποίο σημαίνει ότι το i δεν είναι στο δένδρο, τερματίζουμε την access κάνοντας splaying στον τελευταίο μη κενό κόμβο που προσπελάστηκε κατά την διάρκεια της αναζήτησης (τον κόμβο από τον οποίο η αναζήτηση πετάχτηκε έξω από τον πάτο του δένδρου) και επιστρέφουμε ένα δείκτη στον κενό κόμβο. Αν το δένδρο είναι άδειο, παραλείπουμε την splaying πράξη. (Βλέπε Εικόνα 6.)

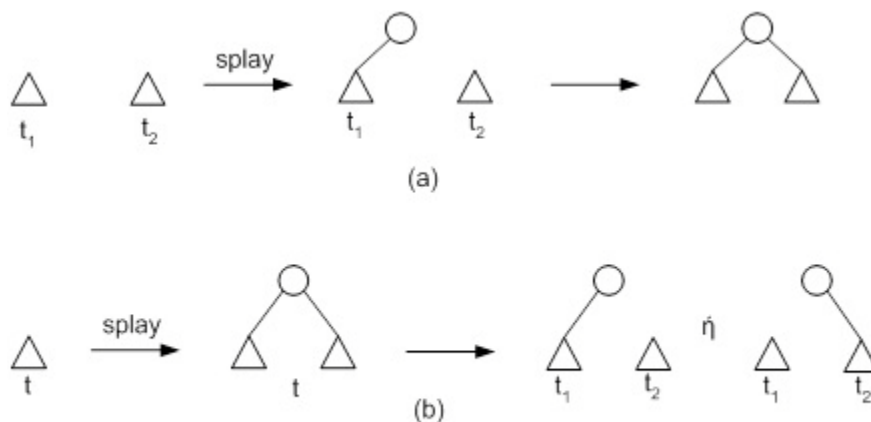


Ε ι κ ο ν α 5 Το Splaying μετά από ανεπιτυχή προσπέλαση του στοιχείου 80

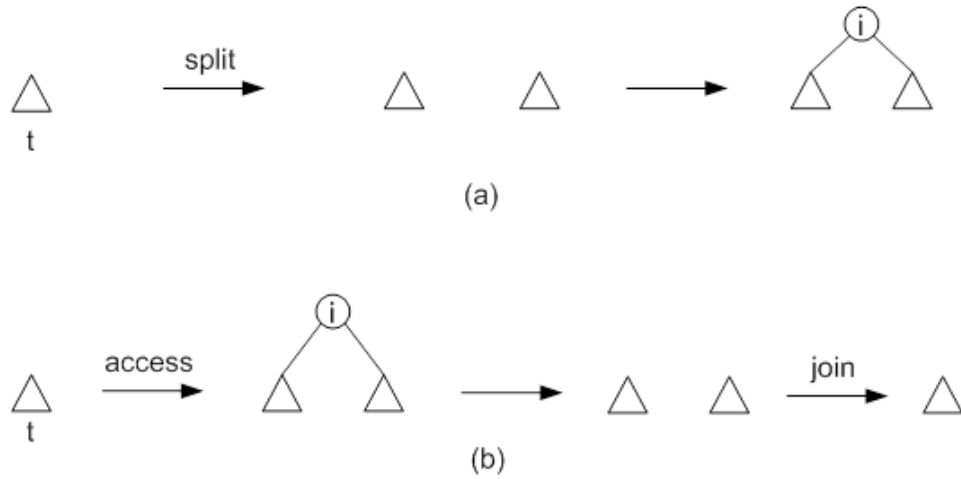
Η μετακίνηση ενός καθορισμένου κόμβου στην ρίζα, σαν αποτέλεσμα του splaying, απλοποιεί σημαντικά την ανανέωση των splay δένδρων. Είναι βολική η υλοποίηση των

πράξεων insert και delete χρησιμοποιώντας τις join και split. Για την επίτευξη της $\text{join}(t_1, t_2)$, ξεκινάμε προσπελαύνοντας το μεγαλύτερο αντικείμενο, έστω i , στο t_1 . Μετά την προσπέλαση, η ρίζα του t_1 περιέχει το αντικείμενο i και για αυτό έχει κενό δεξί παιδί. Συμπληρώνουμε την join κάνοντας το t_2 δεξί υποδένδρο αυτής της ρίζας και επιστρέφοντας το δένδρο που προκύπτει.

Για την πραγματοποίηση της $\text{split}(i, t)$, εκτελούμε πρώτα την $\text{access}(i, t)$ και στη συνέχεια επιστρέφουμε τα δύο δένδρα που σχηματίζονται σπάζοντας είτε τον αριστερό σύνδεσμο είτε τον δεξιό σύνδεσμο της νέας ρίζας του t , ανάλογα με το αν η ρίζα περιέχει αντικείμενο μεγαλύτερο του i ή αντικείμενο όχι μεγαλύτερο του i . (Βλέπε Εικόνα 10.) Στις join και split πρέπει να χειριστεί ειδικά η περίπτωση που η είσοδος της πράξης είναι ένα κενό δένδρο (ή δένδρα).

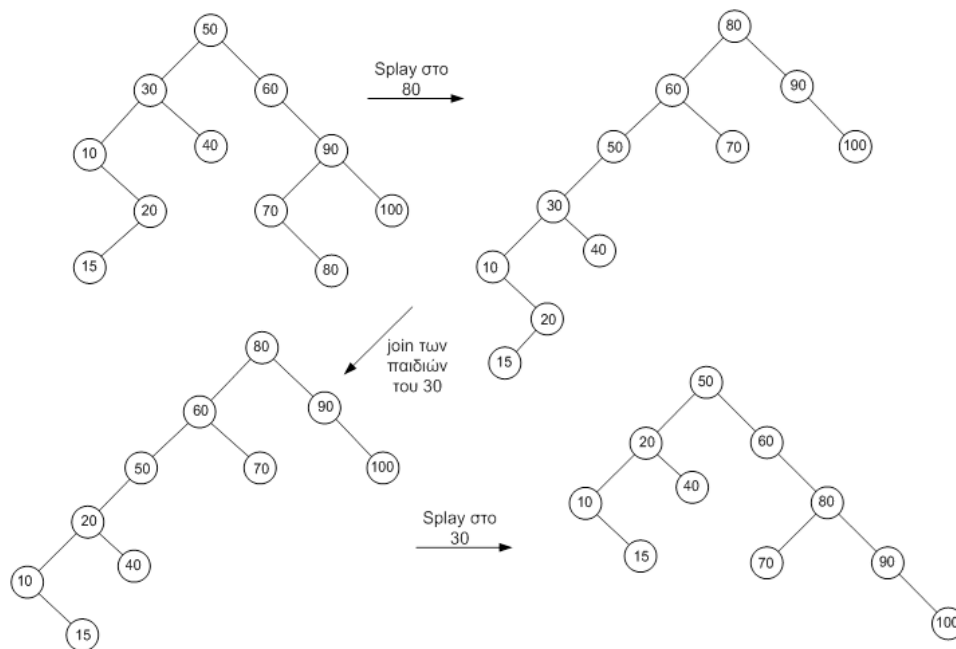


Εικόνα 6 Υλοποίηση της join και split: (a) $\text{join}(t_1, t_2)$ (b) $\text{split}(i, t)$.



Εικόνα 7 Υλοποίηση της insertion και delete με χρήση των join και split: (a) insert(i, t) (b) delete(i, t)

Για την insert(i, t), εκτελούμε την split(i, t) και στη συνέχεια αντικαθιστούμε το t από ένα δένδρο που αποτελείται από μια καινούρια ρίζα που περιέχει το i , του οποίου το αριστερό και το δεξί υποδένδρο είναι τα δένδρα t_1 και t_2 που επιστρέφονται από την split. Για την delete(i, t), εκτελούμε την πράξη access(i, t) και στη συνέχεια αντικαθιστούμε το t από την join πράξη του αριστερού και του δεξιού υποδένδρου του (χωρίς τη ρίζα του t). (Βλέπε Εικόνα 7.)



Εικόνα 8 Εναλλακτική υλοποίηση της insertion και delete. Εισαγωγή του 80 και στη συνέχεια διαγραφή του 30.

Υπάρχουν εναλλακτικές υλοποιήσεις των πράξεων εισαγωγής και διαγραφής (insert και delete) οι οποίες έχουν ελαφρώς καλύτερα όρια επιμερισμένου χρόνου. Για την πραγματοποίηση της $\text{insert}(i, t)$, αναζητούμε το i , στη συνέχεια αντικαθιστούμε τον δείκτη στον κενό κόμβο στον οποίο καταλήξαμε κατά την αναζήτηση με έναν δείκτη σε έναν νέο κόμβο που περιέχει το i , και τέλος κάνουμε splaying στο δένδρο στον καινούριο κόμβο. Για την $\text{delete}(i, t)$, αναζητούμε τον κόμβο που περιέχει το i . Έστω x αυτός ο κόμβος και έστω y ο πατέρας του. Αντικαθιστούμε τον κόμβο x από παιδί του y με την join του αριστερού και του δεξιού υποδένδρου του x , και στη συνέχεια εφαρμόζουμε splaying στον κόμβο y . (Βλέπε Εικόνα 8.)

Λήμμα ανανέωσης (update lemma). Οι επιμερισμένοι χρόνοι των πράξεων του splay δένδρου έχουν τα ακόλουθα άνω όρια, με W να είναι το συνολικό βάρος των αντικειμένων στο δένδρο ή στα δένδρα που χρησιμοποιούνται για την πράξη:

$$\begin{aligned}
\text{access}(i, t): & \begin{cases} 3\log\left(\frac{W}{w(i)}\right)+1 & \text{αν } i = t \in \\ 3\log\left(\frac{W}{\min\{w(i-), w(i+)\}}\right)+1 & \text{αν } i \neq t \notin \end{cases} \\
\text{join}(t_1, t_2): & 3\log\left(\frac{W}{w(i)}\right) + O(1), \text{ όπου } i \text{ είναι το τελευταίο αντικείμενο στο } t_1 \\
\text{split}(i, t): & \begin{cases} 3\log\left(\frac{W}{w(i)}\right) + O(1) & \text{αν } i = t \in \\ 3\log\left(\frac{W}{\min\{w(i-), w(i+)\}}\right) + O(1) & \text{αν } i \neq t \notin \end{cases} \\
\text{insert}(i, t): & 3\log\left(\frac{W - w(i)}{\min\{w(i-), w(i+)\}}\right) + \log\left(\frac{W}{w(i)}\right) + O(1) \\
\text{delete}(i, t): & 3\log\left(\frac{W}{w(i)}\right) + 3\log\left(\frac{W - w(i)}{w(i-)}\right) + O(1).
\end{aligned}$$

2.2 Πώς Εφαρμόζουμε Splay για LogLogN-ανταγωνιστικότητα

Ο George F. Georgakopoulos [2] παρουσιάζει μια επέκταση της splay τεχνικής, τη splay-αλυσίδα (chain-splay). Τα δέντρα chain-splay ανεβάζουν το προσεγγισμένο στοιχείο στη ρίζα ακριβώς όπως τα κλασικά splay δέντρα, αλλά εκτελούν μερικά splay τοπικού χαρακτήρα κάτω από το προσεγγισμένο στοιχείο. Αποδεικνύεται πως το Chain-Splay είναι $\log\log n$ - ανταγωνιστικό σε οποιοδήποτε σε μη απευθείας σύνδεση αλγόριθμο αναζήτησης. Αυτό το αποτέλεσμα είναι το κοντινότερο στην εικασία δυναμικής βελτιστότητας των splay δέντρων από το 1983.

2.2.1 Εφαρμόζοντας splaying για να επιτύχουμε το χαμηλότερο όριο μέσα σε έναν παράγοντα $O(\log\log N)$

Για να εξυπηρετηθεί μία ακολουθία προσπελάσεων H , πρέπει σε κάθε βήμα i να προσπελαυνεται το στοιχείο s_i και να μετατρέπεται το δέντρο S_i στο δέντρο S_{i+1} . Για την επιμερισμένη ανάλυση κόστους αυτών των προσβάσεων καθορίζουμε το ατομικό βάρος $W_i(x)$ του x στο S_i ως:

$$w_i(x) = (\log 2(N+1))^{-\text{level}_i(x)},$$

το level (επίπεδο) του i προέρχεται από τον αντίστοιχο κόμβο του B και την τρέχουσα διαμέριση P_i .

Λήμμα: Έστω το δέντρο S με ρίζα r ζυγισμένο σύμφωνα με το $w(\cdot)$. Τότε το επιμερισμένο κόστος για την πρόσβαση σε οποιουδήποτε x στο S είναι

$$O(\varphi(x \rightarrow r)) = O(\log \log 2(N+1)) \cdot \text{level}(x).$$

2.2.2 Συνθήκη σεβασμού αλυσίδας

(1) Για κάθε βήμα $i = 0, \dots, M-1$, κάθε αλυσίδα C στη προκληθείσα διαμέριση στο B από την ιστορία H μέχρι το βήμα i , διαμορφώνει ένα ενιαίο υποδένδρο στο S_i (δηλ. οι κόμβοι του συνδέονται με το S_i) και ο κόμβος γονέας του υψηλότερου κόμβου του C στο S_i ανήκει στο γονέα του C στο B .

(2) Σε κάθε κόμβο x στο S διατηρείται μία μεταβλητή που μας δείχνει εάν ο x είναι ο κορυφαίος κόμβος της αλυσίδας στην οποία ανήκει. Είναι πολύ εύκολο να διατηρηθεί αυτή η μεταβλητή κατά τη διάρκεια των περιστροφών. Διατηρείται επίσης το βάθος, $d(x)$, στο B του αντίστοιχου κόμβου στο S . Για αυτό χρειάζονται ακριβώς $\log \log 2(N+1)$ κανονικοποιημένα bits.

1. Chain-Splay(tree S , element σ):

//Splay σ to next higher 'top' node

// unless σ //is already one

2. while $\sigma \neq \text{root}(S)$ do

//Splay σ to the next higher „top“ node

3. $xL = \text{left}(\sigma)$, $xR = \text{right}(\sigma)$

4. If $d(xL) < d(xR)$ then LeftGroup(xL)

5, else RightGroup(xR)

// where d is depth

// ReWeight S - now!

6. End while

7. End Procedure

procedure LeftGroup(x)

1. $cJ \leftarrow \text{NIL}$; $z \leftarrow \text{left}(x)$

```

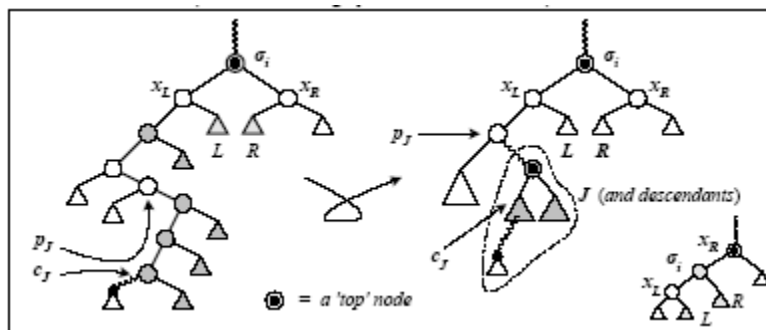
2. while (z ≠ NIL) and (z not a „top“ node) do
3. cJ ← z;
4. if d(z) < d(x) then z ← right(z)
5. else z ← left(z)
6. End while
7. pJ ← the predecessor of cJ along the traversed path
8. Splay cJ to below pJ
9. Splay pJ to below x
10. Mark right(pJ) as „top“

```

End Procedure

procedure RightGroup(x)

// RightGroup(x) is symmetrical



Εικόνα 9 Η LeftGroup πράξη

Θεώρημα: Το Chain-Splay είναι $O(\log \log 2(N+1))$ - ανταγωνιστικό στο βέλτιστο αλγόριθμο αναζήτησης.

2.2.3 Chain Splay με πολυπλοκότητα χειρότερης περίπτωσης $O(\log n)$

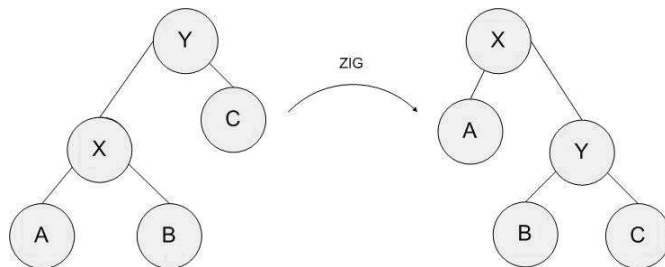
Σε αυτήν την ενότητα θα αποδειχθεί πως το Chain Splay δέντρο έχει πολυπλοκότητα χειρότερης περίπτωσης $O(\log n)$ ανά ερώτημα. Όπως αναφέρει και ο Demaine et al. [4], το δέντρο Tango δύναται να επιτύχει $O(\log n)$ πολυπλοκότητα χειρότερης περίπτωσης ανά πράξη μέσω της αντικατάστασης της βοηθητικής δενδρικής δομής από ένα ισοζυγισμένο BST που υποστηρίζει τις πράξεις της αναζήτησης, διάσπασης και συνένωσης σε $O(1 + \log r)$ χρόνο χειρότερης περίπτωσης, όπου r ισούται με 1 συν τη διαφορά τάξης μεταξύ του στοιχείου προς προσπέλαση και του αμέσως προηγούμενως προσπελασμένου στοιχείου. Θα αποδείξουμε πως και τα Chain - Splay δέντρα έχουν $O(\log n)$ πολυπλοκότητα χειρότερης περίπτωσης. Για να το επιτύχουμε αυτό, θα

αντικαταστήσουμε το δυαδικό δέντρο της δομής με ένα biased 2-b δέντρο [5] έτσι ώστε οι πράξεις που χρησιμοποιούνται για την εκτέλεση του splay σε αυτήν την περίπτωση να οδηγούν σε ένα συνολικό κόστος τάξης $O(\log n)$ στη χειρότερη περίπτωση.

Θεώρημα : Το Chain - Splay δέντρο δύναται να επιτύχει $O(\log n)$ πολυπλοκότητα χρόνου στη χειρότερη περίπτωση ανά πράξη.

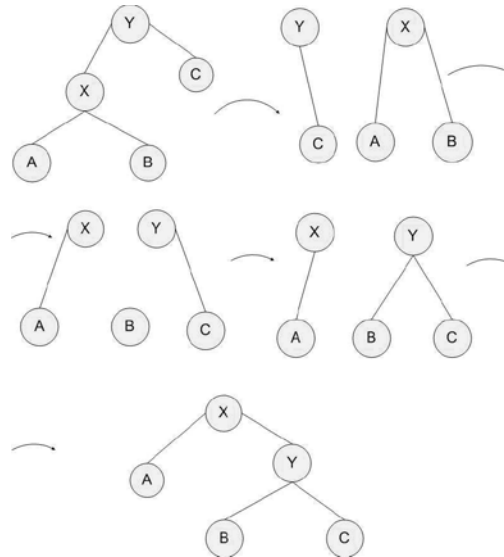
Απόδειξη:

Αρχικά, θα αποδειχθεί πως οι βασικές πράξεις σε ένα splay δέντρο, δηλαδή οι περιστροφές που χρησιμοποιούνται για την εφαρμογή του splay, μπορούν να υλοποιηθούν μέσω της χρήσης των πράξεων συνένωσης και διάσπασης που υποστηρίζονται από ένα biased δέντρο αναζήτησης. Στο επόμενο παράδειγμα, υλοποιείται η περίπτωση μιας απλής περιστροφής με τις πράξεις συνένωσης και διάσπασης. Ας υποθέσουμε πως εκτελούμε μια δεξιά περιστροφή στον κόμβο y στο δέντρο T (Εικόνα 10).



Εικόνα 10 Δεξιά Περιστροφή

Αρχικά, εκτελείται διάσπαση στον κόμβο y ($\text{split}(y)$) αφαιρώντας τον από το υπόλοιπο δέντρο. Στη συνέχεια, εκτελείται $\text{split}(x)$. Η δομή σε αυτή την περίπτωση είναι αυτή που φαίνεται στην Εικόνα 11. Ακολουθώς, πρέπει να διασπαστεί το υποδέντρο B από το x . Η επόμενη πράξη που εκτελείται είναι η συνένωση $\text{catenate}(y, B)$ και μετά η $\text{catenate}(x, y)$. Τέλος, συνενώνεται ο κόμβος x με το υπόλοιπο δέντρο. Συνεπώς, το δέντρο που προκύπτει είναι το επιθυμητό. Η διαδικασία για την αριστερή περιστροφή είναι συμμετρική.



Εικόνα 11 Μία απλή περιστροφή με τις πράξεις concatenate και split

Όσον αφορά την πολυπλοκότητα των πράξεων, στα biased δέντρα αναζήτησης η πράξη της διάσπασης $\text{split}(u)$ κοστίζει σε χρόνο $O(\text{rank}(p) - \text{rank}(u))$, όπου η πράξη $p = \text{path}(u)$ επιστρέφει το μονοπάτι που περιέχει το u και $\text{rank}(p) = \text{rank}(\text{head}(p)) = \text{rank}(\text{ρίζα του δέντρου})$. Η πράξη της συνένωσης $\text{concatenate}(\text{path } p, q)$ απαιτεί $O(\max\{\text{rank}(p), \text{rank}(q)\} - \max\{\text{rank}(u), \text{rank}(w)\})$, όπου $u = \text{tail}(p)$ και $w = \text{head}(q)$. Κατά συνέπεια, το κόστος για μία περιστροφή συγκεντρωτικά ισούται με $\text{rank}(p) - \text{rank}(y) + \text{rank}(y) - \text{rank}(x) + \text{rank}(x) - \text{rank}(B) + \text{rank}(y) - \text{rank}(B) + \text{rank}(x) - \text{rank}(y) + \text{rank}(p) - \text{rank}(x) = 2\{\text{rank}(p) - \text{rank}(B)\}$. Επομένως, για την υλοποίηση του splay, εκτελούμε μία σειρά από περιστροφές σε έναν κόμβο μέχρι να φτάσει στη ρίζα.

Αθροίζοντας τα κόστη, το συνολικό κόστος της πράξης splay είναι τάξεως $\text{rank}(\text{ρίζας}) - \text{rank}(\text{φύλλο})$ ή $O(\log n)$ στη χειρότερη περίπτωση. Όσον αφορά τις τοπικές επαναζυγιστικές πράξεις LeftGroup και RightGroup, και αυτές παρομοίως αποτελούνται από μία σειρά περιστροφών, εφόσον εκτελούνται splay πράξεις στον κόμβο του υποδέντρου. Συγκεκριμένα, το κόστος του Splay σε αυτήν την περίπτωση είναι τάξεως $\text{rank}(\text{ρίζα υποδέντρου}) - \text{rank}(\text{κόμβος υπό splay})$ ή το πολύ $O(\log n)$. Κατά συνέπεια, η πολυπλοκότητα υλοποίησής τους είναι επίσης ίδιας τάξεως και ικανοποιεί τον προαναφερθέν συνολικό όριο. Επομένως το θεώρημα ισχύει.

2.2.4 Τυχαιοποιημένο Chain Splay

Το επιμερισμένο κόστος για την πρόσβαση σε οποιουδήποτε x στο δένδρο S είναι $O(\phi(x \rightarrow r)) = O(\log \log 2(N+1)) \cdot \text{level}(x)$. Η τυχαιοποίηση παρεμβάλλεται στις περιπτώσεις όπου υπάρχει αλλαγή προτιμώμενου παιδιού, ώστε με πιθανότητα p να εφαρμόζεται chain-splay και με πιθανότητα $1-p$ το δέντρο παραμένει ανέγγιχτο. Επομένως, εφόσον ικανοποιείται το Interleave bound και η πιθανότητα p εξασφαλίζει πως ο αριθμός των εφαρμογών του chain-splay είναι το πολύ ίσος με το Interleave bound, η τυχαιοποιημένη έκδοση της τεχνικής είναι $O(\log \log N)$ ανταγωνιστική προς την κλασική δομή.

2.3 Εφαρμογή των Splay Δέντρων στη Συμπίεση Δεδομένων

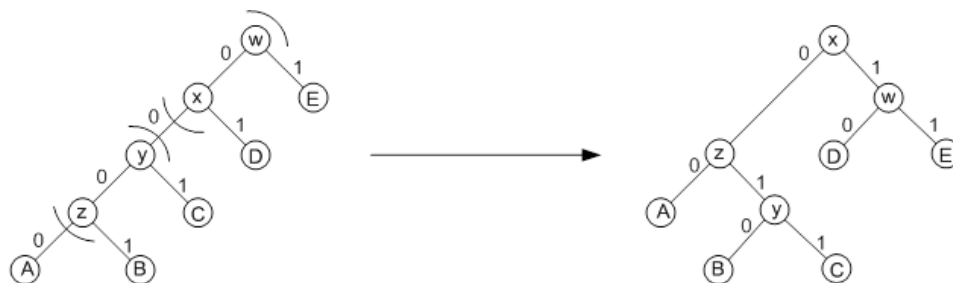
Αν και υπάρχουν διάφορες ειδικές προσεγγίσεις στη συμπίεση δεδομένων, παραδείγματος χάριν run-length κωδικοποίηση, υπάρχουν επίσης διάφορες συστηματικές προσεγγίσεις. Οι κώδικες Huffman είναι μεταξύ των παλαιότερων συστηματικών προσεγγίσεων στη συμπίεση δεδομένων. Οι προσαρμοζόμενοι αλγόριθμοι συμπίεσης Huffman απαιτούν τη χρήση σχημάτων ισοζυγισμένων δένδρων που μπορούν επίσης να εφαρμοστούν στις δομές δεδομένων που απαιτούνται από τους προσαρμοζόμενους αριθμητικούς αλγορίθμους συμπίεσης. Υπάρχει ικανοποιητική ομοιότητα μεταξύ των ισοζυγισμένων αντικειμένων αυτών των σχημάτων και εκείνων που επιτυγχάνονται από τα splay δέντρα, για να δοκιμαστούν τα splay δέντρα και στα δύο πλαίσια.

Τα splay δέντρα θεωρούνται συνήθως μορφές λεξικογραφικά διαταγμένων δένδρων αναζήτησης, αλλά τα δέντρα που χρησιμοποιούνται στη συμπίεση δεδομένων δεν χρειάζεται να έχουν στατική διάταξη. Η αφαίρεση του περιορισμού διάταξης επιτρέπει στις βασικές πράξεις splaying να απλοποιηθούν αρκετά. Οι προκύπτοντες αλγόριθμοι είναι εξαιρετικά γρήγοροι και συμπαγείς. Όταν εφαρμόζεται σε κώδικες Huffman, το splaying οδηγεί σε έναν τοπικά προσαρμοζόμενο αλγόριθμο συμπίεσης ο οποίος είναι εντυπωσιακά απλός καθώς επίσης και γρήγορος, αν και δεν επιτυγχάνει τη βέλτιστη συμπίεση. Όταν εφαρμόζεται σε αριθμητικούς κώδικες, το αποτέλεσμα είναι σχεδόν βέλτιστο σε συμπίεση και ασυμπτωτικά βέλτιστο σε χρόνο.

Οι περισσότεροι μελετημένοι αλγόριθμοι συμπίεσης δεδομένων είναι βασισμένοι στους κώδικες Huffman. Σε έναν κώδικα Huffman, κάθε χαρακτήρας πηγής αναπαριστάται στο συμπιεσμένο κείμενο από έναν κώδικα μεταβλητού μήκους. Οι κοινοί χαρακτήρες πηγής αναπαριστώνται από σύντομους σε μήκος κώδικες, ενώ οι ασυνήθιστοι χαρακτήρες από μεγάλους σε μήκος κώδικες. Οι κώδικες που χρησιμοποιούνται στο συμπιεσμένο κείμενο πρέπει να υπακούουν στην ιδιότητα προθέματος, η οποία λέει ότι ο κώδικας που χρησιμοποιείται σε συμπιεσμένο κείμενο δεν μπορεί να αποτελεί πρόθεμα οποιουδήποτε άλλου κώδικα.

Οι κώδικες προθέματος μπορούν να θεωρηθούν ως δέντρα, στα οποία κάθε φύλλο σχετίζεται με έναν χαρακτήρα του πηγαίου αλφάβητου. Δεδομένου ότι ένα δένδρο Huffman είναι μία περίπτωση ενός στατικού ισοζυγισμένου δένδρου, αυτό σημαίνει ότι το splaying θα μπορούσε να εφαρμοστεί στη συμπίεση δεδομένων, και ότι ο συμπιεσμένος κώδικας που προκύπτει ως αποτέλεσμα ενός splay δένδρου προθεματικού κώδικα θα είναι, κατά έναν σταθερό παράγοντα, ανάλογος του μεγέθους που επιτυγχάνεται χρησιμοποιώντας κώδικα Huffman.

Όπως περιγράφηκε προηγουμένως, το splaying εφαρμόζεται σε δέντρα όπου τα δεδομένα αποθηκεύονται σε εσωτερικούς κόμβους και όχι στα φύλλα. Τα δέντρα προθεματικού κώδικα μεταφέρουν όλα τα δεδομένα τους στα φύλλα, και τίποτα στους εσωτερικούς κόμβους. Υπάρχει μια παραλλαγή του splaying, που ονομάζεται semi-splaying, η οποία μπορεί να εφαρμοστεί στα δέντρα προθεματικού κώδικα. Στο semi-splaying, ο κόμβος-στόχος (έτσι θα ονομάζουμε από εδώ και πέρα τον κόμβο που είναι να προσπελαστεί) δεν κινείται προς τη ρίζα, ούτε τα παιδιά του τροποποιούνται, αλλά αυτό που συμβαίνει είναι ότι το μήκος του μονοπατιού από τη ρίζα στον κόμβο-στόχο υποδιπλασιάζεται σε μήκος. Το semi-splaying έχει αποδειχθεί ότι επιτυγχάνει τα ίδια θεωρητικά όρια με το splaying, κατά έναν σταθερό παράγοντα.



Εικόνα 12 Semi-splaying στον αριστερότερο κόμβο-φύλλο σε ένα δένδρο κώδικα

2.4 Ο Αλγόριθμος Splay-Προθέματος

Ο αλγόριθμος που θα παρουσιαστεί σε αυτό το σημείο ανήκει στον Douglas W. Jones[3]. Οι δομές δεδομένων που χρειάζονται για τον κώδικα αυτό, θα κατασκευαστούν χρησιμοποιώντας μόνο πίνακες (arrays), αν και η λογική δομή θα μπορούσε να εκφραστεί σαφέστερα με τη χρήση εγγραφών (records) και δεικτών (pointers). Κάθε εσωτερικός κόμβος στο δένδρο κώδικα πρέπει να επιτρέπει προσπέλαση στα δύο παιδιά και στον πατέρα του κόμβου. Ο ευκολότερος τρόπος για να γίνει αυτό είναι με τη χρήση τριών δεικτών ανά κόμβο.

Αφού κάθε χαρακτήρας συμπίεστεί ή αποσυμπίεστεί, χρησιμοποιώντας αυτή την εκδοχή του δένδρου κώδικα, στο δένδρο πρέπει να γίνει splay στον κώδικα αυτού του χαρακτήρα. Για την συμπίεση ενός χαρακτήρα πηγαίου κειμένου, ο χαρακτήρας θα πρέπει να κωδικοποιηθεί χρησιμοποιώντας το δένδρο κώδικα, και στη συνέχεια να μεταδοθεί. Αφού γίνει η κωδικοποίηση ακολουθώντας ένα μονοπάτι από το φύλλο στη ρίζα του δένδρου, τα bits του κώδικα παράγονται με την αντίστροφη σειρά από τη σειρά με την οποία πρέπει να μεταδοθούν. Για την αποσυμπίεση ενός χαρακτήρα, τα επιτυχημένα bits πρέπει να διαβαστούν από το συμπίεσμένο κείμενο χρησιμοποιώντας τη συνάρτηση receive. Κάθε bit καθορίζει ένα βήμα στο μονοπάτι από τη ρίζα του δένδρου στο φύλλο που αναπαριστά τον αποσυμπίεσμένο χαρακτήρα.

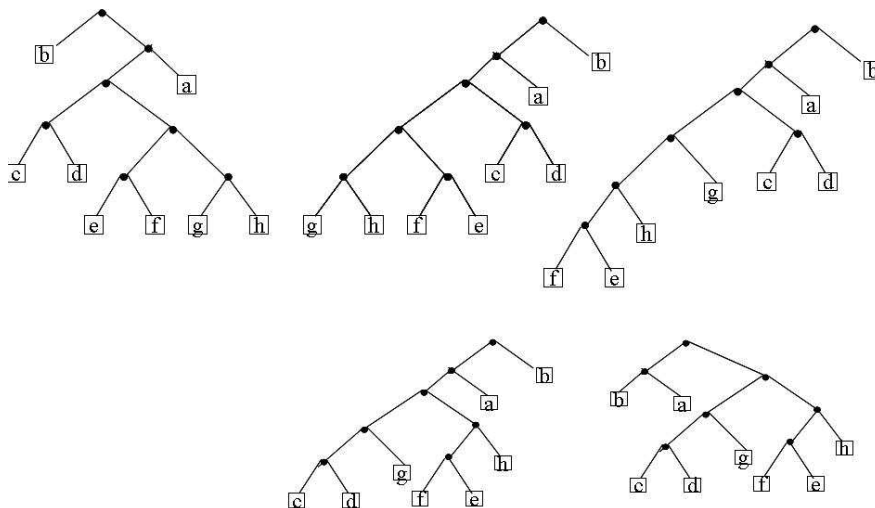
Ο αλγόριθμος λειτουργεί βηματικά ως εξής: Έστω ότι επιθυμούμε να συμπίεσουμε κείμενο με βάση ένα αλφάβητο Σ . Δημιουργούμε ένα δυαδικό δέντρο που περιέχει τα στοιχεία του Σ στα φύλλα του.

Για να κωδικοποιήσουμε ένα σύμβολο x :

- Διασχίζουμε το μονοπάτι από τη ρίζα στο x επιστρέφοντας 0 όταν κατευθυνόμαστε αριστερά και 1 όταν κατευθυνόμαστε δεξιά.
- Στρέφουμε το δέντρο ώστε το κωδικοποιημένο σύμβολο να βρίσκεται στο πιο αριστερό φύλλο.
- Εφαρμόζουμε Semi-splay στο πιο αριστερό φύλλο (εξαλείφουμε την ανάγκη για την περίπτωση του zig-zag).

- Κατά τη διάρκεια της splay πράξης εκτελούμε ημιπεριστροφές αντί για πλήρεις περιστροφές.
- Χρησιμοποιούμε το νέο δέντρο για να κωδικοποιήσουμε το επόμενο σύμβολο.

Ένα παράδειγμα των μετασχηματισμών του δέντρου κατά τη διάρκεια της διαδικασίας κωδικοποίησης απεικονίζεται στην επόμενη εικόνα. Εδώ, τελικά τα σύμβολα {a,b,g} βρίσκονται πιο κοντά στη ρίζα.



Εικόνα 13 Συμπίεση μέσω splay δέντρων της λέξης "aabg".

2.5 Συμπιέζοντας Χρησιμοποιώντας την Τεχνική του Chain Splay

Στα πλαίσια αυτού του κεφαλαίου θα υλοποιήσουμε τον αλγόριθμο συμπίεσης του Jones [3] χρησιμοποιώντας την τεχνική chain splay. Ο συνδυασμός αυτών των δύο ιδεών θα αποτελέσει και τη θεωρητική βάση των πειραμάτων που θα περιγραφεί αργότερα. Πιο συγκεκριμένα, αναπτύχθηκε μία παραλλαγή της τεχνικής chain splay. Τα δέντρα έχουν υλοποιηθεί χρησιμοποιώντας πίνακες. Χρησιμοποιήθηκαν 4 πίνακες, οι πίνακες Parent και Top με 514 στοιχεία και οι πίνακες Left και Right των απογόνων με 257 στοιχεία. Η ανάγκη για τη διατήρηση πληροφορίας σχετικά με το ποιος κόμβος είναι top στο μονοπάτι [2] από τον προσπελασμένο κόμβο προς τη ρίζα, έχει σαν αποτέλεσμα τη χρήση ενός επιπλέον πίνακα με 514 στοιχεία. Λόγω αυτού του γεγονότος, προκύπτει η αναγκαιότητα για χρήση επιπλέον αποθηκευτικού χώρου σε σχέση με την προσέγγιση του Jones [3]. Εφόσον κατά τη διάρκεια της συμπίεσης παράγονται προθεματικοί κώδικες, για την εφαρμογή της chain splay τεχνικής επιλέχθηκε η bottom up παραλλαγή του splaying, το semi-splaying [1] όπως και στην εργασία του D.W Jones. Αυτή η επιλογή έγινε με βάση το γεγονός ότι δεν ενδιαφερόμαστε για τους εσωτερικούς κόμβους του δέντρου, εφόσον η πληροφορία κρατείται στα φύλλα και δεν πρέπει να αφαιρεθεί από αυτά, και επομένως, με άλλα λόγια, προσπελαύνουμε τον πατέρα του κόμβου/φύλλου που στοχεύουμε.

Η κύρια διαφορά της υλοποίησης αυτής της διατριβής και αυτής του Jones [3] είναι ότι όταν καλείται η συνάρτηση Chain semi-splaying, ελέγχει εάν ο κόμβος είναι top και εάν είναι εκτελεί Chain semi-splaying ή απλό semi-splaying σε κάθε άλλη περίπτωση. Τέλος, οι πράξεις LeftGroup και RightGroup κατά τη διάρκεια της εφαρμογής του chain semi-splaying έχουν υλοποιηθεί όπως περιγράφεται στο [2].

2.6 Πειραματική Σύγκριση των Διαφορετικών Μεθόδων Συμπίεσης.

Σκοπός αυτής της ενότητας είναι η πειραματική σύγκριση του αλγορίθμου συμπίεσης με τη χρήση της τεχνικής Chain-Splay με τον κλασικό αλγόριθμο splay-προθέματος. Παράλληλα συγκρίνονται οι κανονικές εκδόσεις των αλγορίθμων με τις τυχαιοποιημένες εκδόσεις τους. Τέλος, και οι δύο αλγόριθμοι συγκρίνονται με τον αλγόριθμο συμπίεσης Huffman.

Η συμπίεση Huffman [6] μειώνει το μέσο μήκος κώδικα που χρησιμοποιείται για να αναπαραστήσει τα σύμβολα της αλφαβήτου. Τα σύμβολα του αλφάβητου που συναντώνται συχνά αποδίδονται με κωδικούς σύντομου μήκους. Η γενική στρατηγική είναι να καταστεί δυνατή η διαφοροποίηση του μήκους του κωδικού από χαρακτήρα σε χαρακτήρα και να διασφαλιστεί ότι οι συχνότεροι χαρακτήρες έχουν μικρότερους κώδικες.

Η συμπίεση Huffman πραγματοποιείται με την κατασκευή ενός δυαδικού δέντρου, χρησιμοποιώντας ένα απλό σύνολο ως παράδειγμα. Αυτό γίνεται με την ταξινόμηση των συμβόλων του αλφάβητου με φθίνουσα σειρά πιθανότητας εμφάνισης και στη συνέχεια, προσθέτοντας κατ'επανάληψη τις δύο χαμηλότερες πιθανότητες. Η διαδικασία αυτή συνεχίζεται έως ότου το άθροισμα πιθανοτήτων των δύο τελευταίων συμβόλων είναι 1. Μόλις ολοκληρωθεί αυτή η διαδικασία, μπορεί να δημιουργηθεί ένα Huffman δυαδικό δέντρο. Η πιθανότητα 1 που αποτελεί το τελευταίο σύμβολο είναι η ρίζα του δυαδικού δέντρου. Οι προκύπτουσες κωδικές λέξεις σχηματίζονται στη συνέχεια ακολουθώντας τη διαδρομή από τη ρίζα προς τα φύλλα αναθέτοντας 0 και 1 στις διακλαδώσεις.

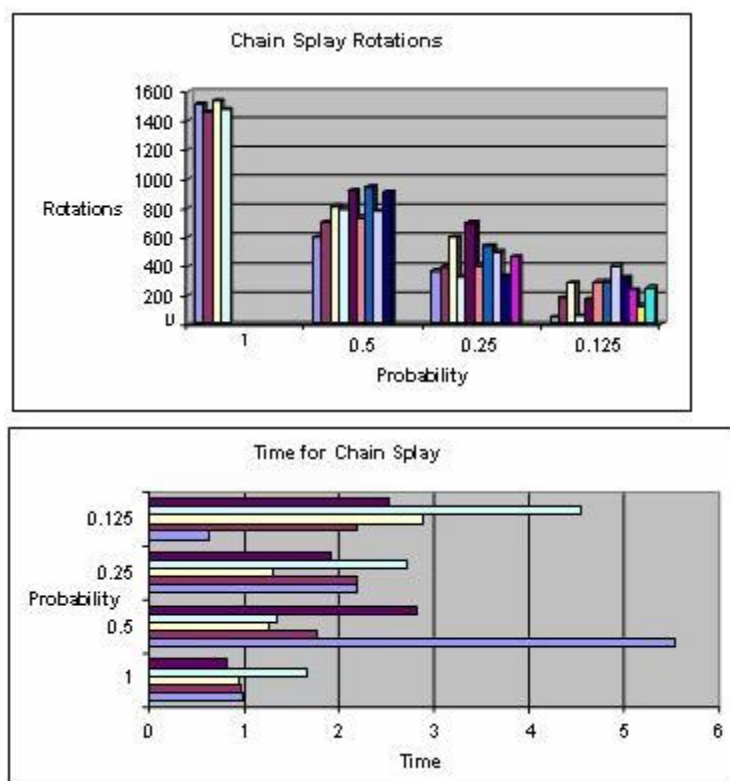
Όσον αφορά συγκεκριμένες λεπτομέρειες για την εφαρμογή του Chain Splay και του τυχαιοποιημένου Chain Splay, ακολουθούν ορισμένες λεπτομέρειες υλοποίησης. Όλες οι δομές δεδομένων υλοποιήθηκαν σε μια κοινή υπολογιστική πλατφόρμα και το λογισμικό υλοποιήθηκε με την αντικειμενοστρεφή γλώσσα προγραμματισμού C++. Πρώτα απ'όλα, επιλέχθηκε το σύνολο στοιχείων στα οποία εκτελέστηκαν τα πειράματα. Οι ακολουθίες αιτήσεων για τα πειράματα παρήχθησαν τυχαία από κατανομές πιθανότητας και δόθηκε έμφαση στην ομοιόμορφη κατανομή ή σε ορισμένες περιπτώσεις με υψηλή τοπικότητα αναφοράς. Πιο συγκεκριμένα, ακολουθίες ακεραίων εισάγονται στα δέντρα και μετρώνται ο χρόνος και οι απαραίτητες πράξεις. Οι τυχαιοποιημένες εκδόσεις των τεχνικών αναμένεται να έχουν καλύτερη απόδοση από ό,τι οι μη τυχαιοποιημένες εκδόσεις ειδικά σε περιπτώσεις όπου η ποσότητα των δεδομένων είναι τόσο μικρή που το splaying είναι περιττό, λόγω της τοπικότητας των στοιχείων. Για παράδειγμα, η πράξη του splay σε αυτές τις περιπτώσεις οδηγεί σε πρόσθετες περιστροφές και καταναλώνεται περισσότερος χρόνος, ενώ το στοιχείο είναι σε θέση να προσπελαστεί σε σχετικά σύντομο χρονικό διάστημα.

Όσον αφορά το χτίσιμο του δέντρου, καθορίζεται η πιθανότητα πρόσβασης του κάθε στοιχείου που έχει εισαχθεί στο δέντρο. Στη συνέχεια, πραγματοποιείται ένας μεγάλος αριθμός πράξεων για να οδηγήσει σε αξιόπιστα και ασφαλή συμπεράσματα. Η απόδοση

των αλγορίθμων αξιολογείται με καταγραφή των συγκρίσεων και των περιστροφών που απαιτούνται για την εκτέλεση της ακολουθίας των πράξεων.

2.7 Σύγκριση των Τυχαιοποιημένων και Μη Τυχαιοποιημένων Εκδόσεων και Συμπεράσματα

Στο Chain Splay, ο αλγόριθμος εκτελείται με πιθανότητα p , 50%, 25% και 12,5%. Γίνεται καταμέτρηση των περιστροφών και του χρόνου που απαιτείται για μια ακολουθία προσβάσεων σε 105 στοιχεία για κάθε πιθανότητα p . Αναμένεται ότι ο αριθμός των περιστροφών που απαιτείται για τον τυχαιοποιημένο αλγόριθμο, να είναι αρκετά μικρότερος από αυτόν που απαιτείται από την ντετερμινιστική έκδοσή του αλγορίθμου. Αυτή η μείωση προκύπτει από το γεγονός ότι σε μια ακολουθία προσβάσεων, ο αλγόριθμος Chain Splay δεν εκτελείται με πιθανότητα $1-p$. Ως εκ τούτου, αποφεύγεται κάποια μερίδα των περιστροφών. Αυτό θα είχε ως συνέπεια, ο ουσιώδης χρόνος για μια ακολουθία προσβάσεων να είναι μικρότερος στις ανωτέρω περιπτώσεις. Παρατηρώντας μια ακολουθία προσβάσεων, επισημαίνεται ότι, όταν η πιθανότητα εκτέλεσης είναι σχετικά μικρή, όπως 12,5%, ο απαιτούμενος χρόνος για την πρόσβαση είναι μεγαλύτερος από τον αντίστοιχο για την πιθανότητα 25%. Αυτό προκύπτει από το γεγονός ότι για $p = 12,5\%$, για μια πρόσβαση σε ένα στοιχείο x σε μεγάλο βάθος, ενώ κανονικά θα εκτελούνταν ο αλγόριθμος, προχωρούμε στην επόμενη πρόσβαση και, στη συνέχεια, αν προσπελάσσει ένα στοιχείο στη γειτονιά του x , πρέπει να αναζητηθεί σε μεγαλύτερο βάθος, μια περίπτωση που θα μπορούσε να αποφευχθεί σε κάθε άλλη περίπτωση. Κατά συνέπεια, αντίθετα προς ό,τι θα αναμενόταν, για μια ακολουθία 1500 προσβάσεων σε σύνολο 10,000 στοιχείων για πιθανότητα 0,125, επιτυγχάνεται μια βελτίωση για την πιθανότητα 0,25 της τάξης του 40%. Το συμπέρασμα που εξάγεται από τις μετρήσεις αυτές είναι ότι, σε περίπτωση εφαρμογής σχετικά μικρών ακολουθιών προσβάσεων σε μια μεγάλη ποσότητα δεδομένων, η τυχαιοποιημένη έκδοσή είναι προτιμότερη. Γενικά, μπορεί να ειπωθεί ότι η τυχαιοποιημένη έκδοση επιτυγχάνει μια μείωση έως και 3% στις περιστροφές, ανάλογα με την πιθανότητα p . Σε γενικές γραμμές, η ίδια συμπεριφορά συναντάται σε μεγάλες και μικρές ακολουθίες προσβάσεων. Αντίστοιχα, για το κλασικό ντετερμινιστικό splay, απαιτούνται κατά μέσο όρο 750 περιστροφές και ο χρόνος κυμαίνεται από 0,656 έως 4,031 δευτερόλεπτα. Σε ακολουθίες προσβάσεων που έχουν παραχθεί για $p = 1/2$ και $p = 1/4$, η τυχαιοποιημένη splay τεχνική επιτυγχάνει βελτιώσεις από 8% έως 12% σε σχέση με το ντετερμινιστικό splay. Για μικρότερες τιμές του p , οι βελτιώσεις είναι ακόμη μεγαλύτερες, χρησιμοποιώντας την πιθανότητα $p = 1/8$, με βελτίωση κατά 25%. Για τις μικρότερες πιθανότητες p , παρατηρείται χαρακτηριστικά και πάλι μια αύξηση του χρόνου της πρόσβασης. Σε ορισμένες ακολουθίες με μικρή τοπικότητα, τα αποτελέσματα είναι αρνητικά. Οι βελτιώσεις σε αυτές τις περιπτώσεις σχεδόν δεν υφίστανται. Όσο αφορά τις περιστροφές, έχουν μειωθεί δραστικά όσο αυξάνεται η πιθανότητα. Για $p = 1/2$, οι περιστροφές είναι περισσότερες σε αρκετές περιπτώσεις, με τις βελτιώσεις όπου προκύπτουν, να αγγίζουν σχεδόν το 20%.



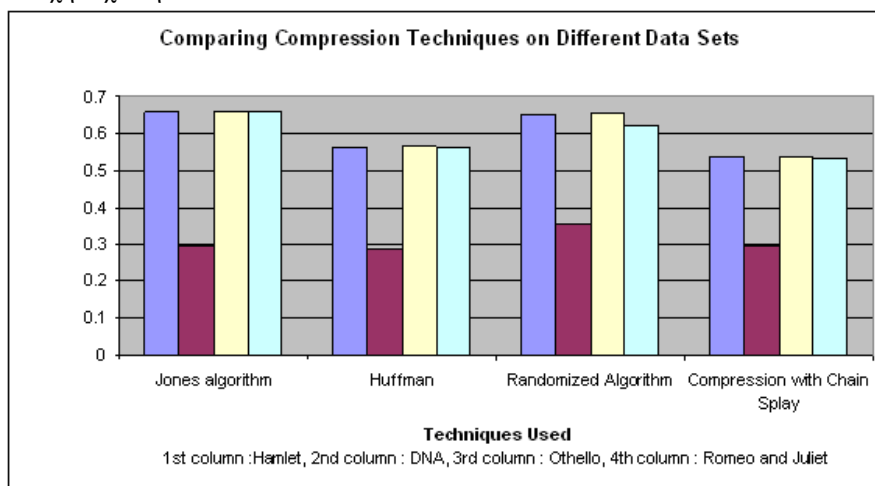
Εικόνα 14 Απόδοση σε χρόνο και περιστροφές που εκτελούνται από τον τυχαιοποιημένο αλγόριθμο για πιθανότητες 100%,50%,25% and 12.5%. Κάθε χρόνο αντιπροσωπεύει μία διαφορετική ακολουθία προσβάσεων.

Όσον αφορά την υλοποίηση του αλγορίθμου συμπίεσης του Jones [3] χρησιμοποιώντας chain-splay, ο πηγαίος κώδικας υλοποιήθηκε σε μια κοινή υπολογιστική πλατφόρμα, όπως τα δέντρα της προηγούμενης ενότητας.

Προκειμένου η εφαρμογή να προσκομίσει τα απαραίτητα πειραματικά συμπεράσματα, βασίστηκε στην εργασία του Jones [3] σχετικά με τη χρήση splay δέντρων για τη συμπίεση τα δεδομένων, ενσωματώνοντας τη Chain Splay τεχνική. Ο συνδυασμός αυτών των δύο ιδεών είναι η θεωρητική βάση των πειραμάτων.

Για τους σκοπούς της αξιολόγησης της συμπίεσης που μπορεί να επιτευχθεί με την ενσωμάτωση του Chain semi-splaying στον Prefix αλγόριθμο, συγκρίνονται οι επιδόσεις του με αυτές των διαφόρων τεχνικών συμπίεσης. Ωστόσο, κύριο μέλημά είναι να αποδειχθεί στην πράξη η υπεροχή του έναντι του κλασικού Prefix αλγορίθμου. Η απόδοση του κλασικού Prefix αλγορίθμου αξιολογήθηκε επίσης χρησιμοποιώντας μια τυχαιοποιημένη εκδοχή του splay αλγορίθμου. Η πιθανότητα p που χρησιμοποιείται είναι

50%. Με άλλα λόγια, η splay τεχνική εφαρμόζεται κάθε φορά με πιθανότητα 50%. Στόχος είναι να εξεταστεί εάν η τυχαιοποιημένη έκδοχή αποδίδει καλύτερα ή χειρότερα από ό,τι η κλασική έκδοχή για δεδομένα έγγραφα. Προκειμένου να ελεγχθεί η αποτελεσματικότητα των αλγορίθμων, έχουν επιλεγεί πολλά έγγραφα σε μορφή txt αλλά και ακολουθίες DNA. Στον παρακάτω πίνακα, παρουσιάζεται η συμπίεση που επιτυγχάνεται από κάθε τεχνική. Τα μεγέθη των εγγράφων μετρώνται σε byte. Στην πρώτη στήλη δηλώνεται το αρχικό μέγεθος του εγγράφου πριν από τη συμπίεση του. Στην επόμενη στήλη, παρουσιάζεται το μέγεθος κάθε συμπιεσμένου εγγράφου από την αντίστοιχη τεχνική.



Εικόνα 15 Σύγκριση διαφορετικών τεχνικών συμπίεσης σε διάφορα σύνολα δεδομένων

Στην Εικόνα 15 Σύγκριση διαφορετικών τεχνικών συμπίεσης σε διάφορα σύνολα δεδομένων παρουσιάζεται το ποσοστό της συμπίεσης που έχει σημειώσει κάθε τεχνική για κάθε έγγραφο. Τα ποσοστά % προκύπτουν από τη διαίρεση του μεγέθους του συμπιεσμένου έγγραφου με εκείνο του πρωτότυπου εγγράφου. Ως εκ τούτου, όσο μικρότερο είναι το ποσοστό, τόσο καλύτερα έχει αποδώσει ο αλγόριθμος. Με βάση αυτό, παρατηρείται ότι η Chain-splay τεχνική έχει την καλύτερη επίδοση μεταξύ όλων (Πίνακας 1). Στην πραγματικότητα, είναι καλύτερη από ό,τι η Huffman κωδικοποίηση στις τρεις από τις τέσσερις περιπτώσεις. Το πιο πολύτιμο αποτέλεσμα είναι το γεγονός ότι η έκδοση του Prefix αλγορίθμου με την Chain-splay τεχνική αποδίδει καλύτερα από τον κλασικό Prefix αλγόριθμο και την τυχαιοποιημένη έκδοσή του. Όσον αφορά τη σύγκριση μεταξύ του κλασικού Prefix αλγορίθμου και της τυχαιοποιημένης έκδοσής του, οι επιδόσεις τους είναι παρόμοιες και για την περίπτωση της πιθανότητας 50% δεν μπορούν να εξαχθούν ασφαλή συμπεράσματα.

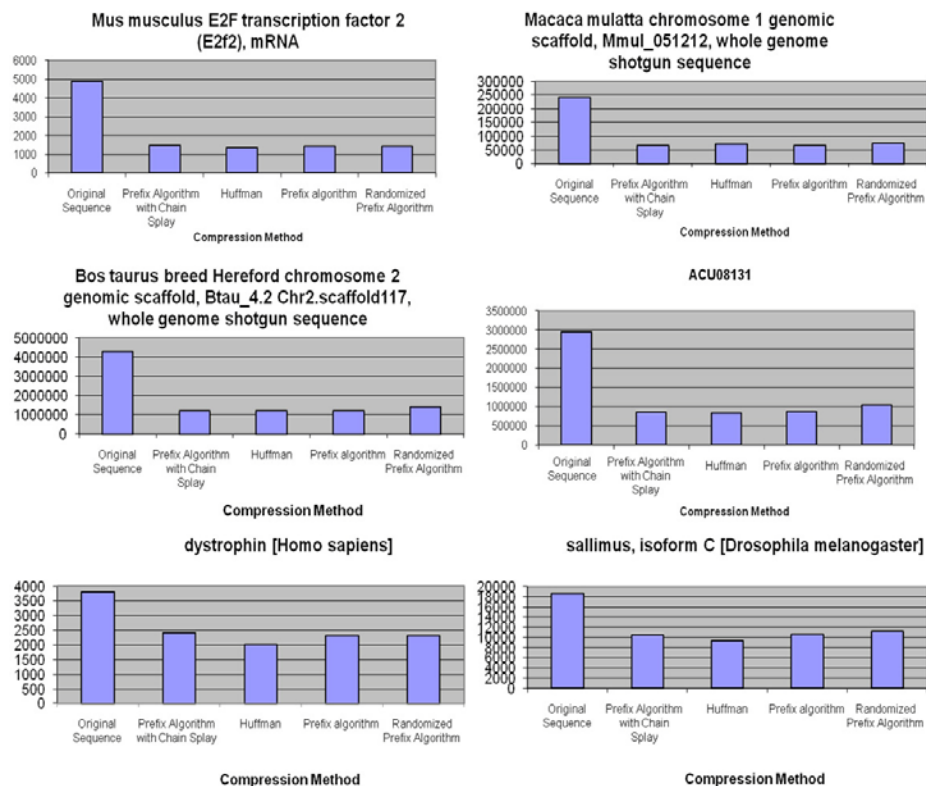
	Original Document	Prefix algorithm	Huffman	Randomized Prefix Algorithm	Prefix Algorithm with Chain Splay
Hamlet	197,254	129,908	111,076	128,451	105,664
DNA	2,946,077	880,551	838,838	1,044,374	860,162
Othello	176,513	116,247	100,020	115,771	94,731
Romeo and Juliet	156,612	103,354	88,084	96,910	83,258

Πίνακας 1

Όσον αφορά τη διαφορά που παρατηρείται όταν αλλάζει η πιθανότητα εκτέλεσης του splay στις τυχαιοποιημένες εκδόσεις, όπως φαίνεται στον πίνακα 2, στις περιπτώσεις των "Othello" και "Hamlet", οι διαφορές μεταξύ των διαφορετικών εκδόσεων είναι μικρής σημασίας, αλλά και πάλι οι τυχαιοποιημένες εκδόσεις έχουν καλύτερες επιδόσεις. Ο κλασικός Prefix αλγόριθμος αποδίδει καλύτερα από όλες τις τυχαιοποιημένες εκδόσεις στη συμπίεση της ακολουθίας του DNA, ενώ η τυχαιοποιημένη έκδοση με πιθανότητα $p = 50\%$ επιτυγχάνει την καλύτερη επίδοση στην περίπτωση του «Romeo and Juliet».

	Prefix algorithm	Randomized Prefix Algorithm with probability $p=50\%$	Randomized Prefix Algorithm with probability $p=25\%$	Randomized Prefix Algorithm with probability $p=12,5\%$
Hamlet	129,908	128,451	129,908	132,654
DNA	880,551	1,044,374	905,946	2,062.91
Othello	116,247	115,771	115,876	157,765
Romeo and Juliet	103,354	96,910	156,622	156,522

Πίνακας 2



Εικόνα 16 Η συμπίεση που επιτυγχάνεται από κάθε τεχνική συμπίεσης σε κάθε βιολογική ακολουθία. Το μέγεθος μετριέται σε bytes.

Προκειμένου να ελεγχθεί η αποτελεσματικότητα των αλγορίθμων, επιλέχθηκαν και διάφορα βιολογικά δεδομένα. Στον πίνακα που ακολουθεί (Πίνακας 3), παρουσιάζεται η συμπίεση που επιτυγχάνεται με κάθε τεχνική. Τα μεγέθη των ακολουθιών μετρώνται και πάλι σε bytes. Στην πρώτη στήλη δηλώνουμε το αρχικό μέγεθος του εγγράφου πριν από τη συμπίεση του και στην επόμενη στήλη, το μέγεθος του συμπιεσμένου εγγράφου από την αντίστοιχη τεχνική.

Όπως βλέπουμε στον Πίνακα 3 και στην Εικόνα 16 όλοι οι αλγόριθμοι αποδίδουν αρκετά καλά επιτυγχάνοντας συμπίεση μεγέθους 25% επί των αρχικών ακολουθιών του DNA, ενώ για την περίπτωση των πρωτεϊνικών ακολουθιών, η συμπίεση είναι περίπου 50%, δεδομένου ότι το αλφάβητο στις πρωτεΐνες αποτελείται από 20 αμινοξέα, αντί των 4 νουκλεοτιδίων του DNA. Όσον αφορά τον Prefix αλγόριθμο με Chain Splay σε σύγκριση με τον Prefix αλγόριθμο, η πρώτη τεχνική αποδίδει καλύτερα στις περισσότερες περιπτώσεις. Πιο συγκεκριμένα, η αύξηση του μεγέθους της ακολουθίας οδηγεί σε ένα καλύτερο ποσοστό συμπίεσης. Στις περιπτώσεις όπου η ακολουθία DNA είναι σχετικά μικρή, όπως το musculus Mus Gene και η δυστροφίνη στον Πίνακα 3, ο κλασικός Prefix αλγόριθμος είναι ελαφρώς καλύτερος. Η εξήγηση για αυτό το γεγονός είναι ότι στις

περιπτώσεις όπου το μέγεθος των δεδομένων είναι μικρό, δεν αξιοποιούνται οι τοπικές πράξεις επανοργάνωσης που εκτελούνται από τον Prefix αλγόριθμο με Chain Splay.

Gene Name	Url	Original Sequence	Prefix Algorithm with Chain Splay	Huffman	Prefix algorithm	Randomized Prefix Algorithm
<i>Mus musculus</i> E2F transcription factor 2 (E2f2), mRNA	http://www.ncbi.nlm.nih.gov/nuccore/160333363?report=fasta	4880	1476	1361	1442	1428
<i>Macaca mulatta</i> chromosome 1	http://www.ncbi.nlm.nih.gov/nuccore/90662813?report=fasta	241909	66233	73094	67582	73969
<i>Bos taurus</i> breed Hereford chromosome 2	http://www.ncbi.nlm.nih.gov/nuccore/109156578?report=fasta	4259901	1216753	1225173	1237067	1421484
ACU08131	http://genome.crg.es/datasets/genomics96/seqs/DNASequences.fasta	2946077	860162	838838	880551	1044374
Protein Name						
dystrophin [Homo sapiens]	http://www.ncbi.nlm.nih.gov/protein/57284025?report=fasta	3789	2421	2005	2322	2312
sallimus, isoform C [Drosophila melanogaster]	http://www.ncbi.nlm.nih.gov/protein/158028409?report=fasta	18659	10548	9370	10658	11234

Πίνακας 3

Οι επιδόσεις του τυχαιοποιημένου Prefix αλγορίθμου δείχνει ότι η τυχαιοποίηση μπορεί να βελτιώσει την επιτυγχανόμενη συμπίεση, μόνο σε ορισμένες περιπτώσεις όπου τα συμπίεσιμα δεδομένα αποτελούνται από πολλές επαναλήψεις των ίδιων προτύπων και δεν υπάρχει καμία ανάγκη να εκτελείται splay σε κάθε πρόσβαση.

Η απόδοση της συμπίεσης Huffman σε σύγκριση με τον Prefix αλγόριθμο με Chain Splay δεν οδηγεί σε ασφαλή συμπεράσματα, αφού και οι δύο τεχνικές αποδίδουν καλύτερα σε ορισμένες περιπτώσεις, όπως φαίνεται στον Πίνακα 3.

Βιβλιογραφία Κεφαλαίου

1. D.D. Sleator and R.E. Tarjan. Self-adjusting binary search trees, Journal of the ACM, 32:652-686, 1985.
2. George F. Georgakopoulos: How to Splay for loglog N-Competitiveness. WEA 2005: 570-579
3. Douglas W. Jones: Application of Splay Trees to Data Compression. Commun. ACM 31(8): 996-1007 (1988)
4. Erik D. Demaine, Dion Harmon, John Iacono, and Mihai P. Dinicai. "Dynamic Optimality—Almost", SIAM Journal on Computing, volume 37, number 1, May 2007, pages 240–251.

5. S. W. Bent, D. D. Sleator and R. E. Tarjan. Biased search trees, SIAM Journal on Computing 14(3), 545-568, 1985.
6. D.A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes", Proceedings of the I.R.E., September 1952, pp 1098-1102.
7. Matthew H. Austern, Bjarne Stroustrup, Mikkel Thorup, John Wilkinson, Untangling the balancing and searching of balanced binary search trees Software: Practice and Experience Volume 33, Issue 13, Date: 10 November 2003, Pages: 1273-129.
8. Jim Bell, Gopal Gupta, An evaluation of self-adjusting binary search tree techniques Software: Practice and Experience Volume 23, Issue 4, Date: April 1993, Pages: 369-382.
9. Gunnar Brinkmann, Jan Degraer, Karel De Loof, Rehabilitation of an unloved child: semi-splaying Software: Practice and Experience Volume 39, Issue 1, Date: January 2009, Pages: 33-45.
10. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition, MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Chapter 12: Binary search trees, pp.253-272. Section 15.5: Optimal binary search trees, pp.356-363.
11. Martin Fürer, Randomized splay trees, Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms, p.903-904, January 17-19, 1999, Baltimore, Maryland, United States.
12. Dennis Grinberg, Sivaramakrishnan Rajagopalan, Ramarathnam Venkatesan, Victor K. Wei: Splay Trees for Data Compression. ,SODA95, 522-530.
13. Eric K. Lee, Charles U. Martel, When to use splay trees Software: Practice and Experience Volume 37, Issue 15, Date: December 2007, Pages: 1559-1575.
14. Alistair Moffat, Gary Eddy, Ola Petersson, Splaysort: Fast, Versatile, Practical Software: Practice and Experience Volume 26, Issue 7, Date: July 1996, Pages: 781-797.
15. C. C. Wang, J. Derryberry, D.D. Sleator, $O(\log \log n)$ - competitive dynamic binary search trees. ACM SODA 2006, 374-383.
16. Hugh E. Williams, Justin Zobel, Steffen Heinz, Self-adjusting trees in practice for large text collections Software: Practice and Experience Volume 31, Issue 10, Date: August 2001, Pages: 925-939.
17. Alistair Moffat, Andrew Turpin: Compression and Coding Algorithms Kluwer 2002.
18. Dorrigiv R., López-Ortiz A. and Munro J. An Application of Self-organizing Data Structures to Compression. 8th Symposium on Experimental Algorithms , 137 - 148, 2009
19. Dimitris Antoniou, Evangelos Theodoridis and Athanasios Tsakalidis, "Compressing Biological Sequences using Self Adjusting Data Structures", The 10th IEEE International Conference on Information Technology and Applications in Biomedicine ,(ITAB 2010).
20. D.Antoniou , I. Kampouris, E.Theodoridis . "Splaying for Compression: A theoretical and Experimental Study", Journal of Discrete Algorithms (JDA) (έχει υποβληθεί).



ΜΕΡΟΣ Γ: Επανοργάνωση Δικτυακών τόπων

Κεφάλαιο 3

Ανάθεση Hotlinks Βασιζόμενη στην Ομοιότητα Περιεχομένου

Σύνοψη

Στο κεφάλαιο αυτό περιγράφεται ένας αποτελεσματικός αλγόριθμος για την επανοργάνωση δικτυακών τόπων. Η ενίσχυση της εμπειρίας της περιήγησης στο διαδίκτυο είναι ένα ανοιχτό ζήτημα που αντιμετωπίζεται συχνά με ανάθεση υπερσυνδέσμων μεταξύ των ιστοσελίδων και συντομεύσεων από τον ένα κόμβο στον άλλο. Στόχος είναι η ανάπτυξη μιας πρωτότυπης, πιο αποτελεσματικής προσέγγισης για την ελαχιστοποίηση του αναμενόμενου αριθμού των βημάτων που απαιτούνται για την πρόσβαση σελίδων κατά την περιήγηση σε ένα δικτυακό τόπο. Παρουσιάζεται ένας τυχαιοποιημένος αλγόριθμος, ο οποίος συνδυάζει τη δημοτικότητα των ιστοσελίδων, τη δομή του δικτυακού τόπου, και για πρώτη φορά την ομοιότητα περιεχομένου μεταξύ των σελίδων, προκειμένου να προτείνει την ανάθεση των κατάλληλων υπερσυνδέσμων. Αποδεικνύεται πειραματικά πως με τη χρήση του αλγορίθμου, οι χρήστες χρειάζονται λιγότερες μεταβάσεις σελίδων για να προσπελάσουν τις επιθυμητές πληροφορίες κατά την περιήγηση σε έναν ιστότοπο, βελτιώνοντας έτσι τη δομή του.

3.1 Εισαγωγή

Ο παγκόσμιος ιστός έχει καταστεί τα τελευταία χρόνια ως μία πλατφόρμα ολοκλήρωσης και ενοποίησης πληθώρας δεδομένων. Τα δεδομένα αυτά πέρα από την αρχική hypertextual μορφή τους, συμπεριλαμβάνουν πλέον κυρίως, streaming audio, βίντεο, και δυναμικό δικτυακό περιεχόμενο. Σημειωτέον, το διαδίκτυο έχει παρουσία και σε φορητές συσκευές, διερύνοντας έτσι το κοινό στο οποίο απευθύνεται.

Οι διάφορες μορφές των μέσων και η διαρκώς επεκτεινόμενη κοινότητα χρηστών του διαδικτύου, θέτουν αρκετές προκλήσεις στους προγραμματιστές και τους διαχειριστές των ιστότοπων, όσον αφορά τη συχνότητα επισκέψεων και την παρουσίαση του περιεχομένου. Ο κύριος στόχος στο σχεδιασμό και την παρουσίαση ενός ιστότοπου είναι η αποδοτική ανάλυση των μοτίβων επισκεψιμότητας των χρηστών και ο ανασχεδιασμός του περιεχομένου του ιστότοπου ώστε να προωθείται το δημοφιλές περιεχόμενο.

Σε αυτό το κεφάλαιο θα επανεξεταστεί η ανάθεση καυτών συνδέσμων (hotlinks), μια δημοφιλής μέθοδος διαμόρφωσης ιστότοπων. Την ιδέα της ανάθεσης hotlinks εισήγαγαν οι Perkowitz and Etzioni [1], σύμφωνα με την οποία προτείνεται η μη καταστροφική διαμόρφωση της δομής των συνδέσμων, με σκοπό την ελαχιστοποίηση του συνολικού μήκους των μονοπατιών προς τις δημοφιλείς σελίδες του ιστότοπου.

Το πρόβλημα δύναται να οριστεί ως εξής. Έστω ένα κατευθυνόμενο δέντρο με ρίζα, $T=(V,E)$ που αναπαριστά μια συλλογή από V ιστοσελίδες/κόμβους που συνδέονται μέσω ενός συνόλου από E συνδέσμους, που όλες είναι προσβάσιμες από μια αρχική ιστοσελίδα, που αναπαρίσταται ως η ρίζα του δέντρου T . Κάθε φύλλο- ιστοσελίδα έχει ένα βάρος αντιπροσωπευτικό της επισκεψιμότητάς της. Με την πρόσθεση των hotlinks, δηλαδή συντομεύσεις από έναν κόμβο σε κάποιον από τους απόγονούς του, ο στόχος είναι η ελαχιστοποίηση του αναμενόμενου αριθμού βημάτων, που απαιτούνται για να επισκεφτεί ο χρήστης τα φύλλα-σελίδες ξεκινώντας την αρχική σελίδα. Έστω T^A το δέντρο το οποίο προκύπτει από την ανάθεση hotlinks A . Ο αναμενόμενος αριθμός βημάτων από τη ρίζα προς ένα φύλλο-ιστοσελίδα δίνεται από τη σχέση

$$E[T^A, p] = \sum_{i_is_a_leaf} d_A(i) p_i \quad (1),$$

όπου $d_A(i)$ είναι η απόσταση του κόμβου i από την ρίζα στο T^A , και $p = \{p_i : i = 1, \dots, n\}$ είναι η πιθανοτική κατανομή στα φύλλα του αρχικού δέντρου T , που ορίζει την κατανομή των βαρών επισκεψιμότητας. Σκοπός είναι η ανεύρεση μιας ανάθεσης A η οποία να ελαχιστοποιεί το $E[T^A, p]$.

Σε αυτή τη διατριβή, το θέμα της ανάθεσης hotlinks μελετάται από μία νέα οπτική γωνία. Παρουσιάζεται ένας αλγόριθμος για να βελτιωθεί η προσβασιμότητα σε δημοφιλείς ιστοσελίδες, προσθέτοντας επιπλέον συνδέσμους σε αυτές από μη δημοφιλείς σελίδες, με τις οποίες όμως έχουν μεγάλη σημασιολογική ομοιότητα. Επιπλέον, στον προτεινόμενο αλγόριθμο, ο σκοπός δεν είναι μόνο η ελαχιστοποίηση μιας μετρικής που χαρακτηρίζει ολόκληρο το website, αλλά η λογική του αλγορίθμου επιδιώκει κυρίως την εφαρμογή μιας τοπικής μετρικής που να καταδεικνύει την προσβασιμότητα κάθε ιστοσελίδας.

Η καινοτομία της πρότασης έγκειται σε 2 σημεία : (i) στην εισαγωγή της έννοιας της ομοιότητας μεταξύ δύο ιστοσελίδων στο μοντέλο της ανάθεσης hotlink και (ii) στην χρησιμοποίηση πιο ρεαλιστικών γραφικών μοντέλων από το κλασσικό μοντέλο δέντρου που χρησιμοποιείται μέχρι και τώρα. Με αυτόν τον τρόπο ο προτεινόμενος αλγόριθμος διαφοροποιείται από τις προηγούμενες εργασίες για την ανάθεση hotlinks. Πιο συγκεκριμένα, σκοπός είναι να εκμεταλλευτούμε για κάθε ιστοσελίδα, την ομοιοτήτά της με τις υπόλοιπες ιστοσελίδες. Υπάρχουν 3 τρόποι με τους οποίους η ομοιότητα μετριέται τυπικά: με βάση το περιεχόμενο, με βάση τους συνδέσμους και με βάση τη χρήση της ιστοσελίδας. Αυτή που μας ενδιαφέρει στη συγκεκριμένη περίπτωση είναι η ομοιότητα που βασίζεται στο περιεχόμενο η οποία υπολογίζεται με βάση τις λέξεις ή τις ακολουθίες λέξεων που εμφανίζονται σε ιστοσελίδες. Έτσι ένας κόμβος θα συνδέεται μέσω ενός hotlink με έναν άλλο κόμβο, όχι μόνο λαμβάνοντας υπόψιν την συχνότητα με την οποία δέχεται επισκέψεις αλλά και την ομοιότητά του με τον άλλον κόμβο.

Για τη διευκόλυνση της παρουσίασης, ακολουθεί κάποια σημειογραφία και κάποιες συμβάσεις που χρησιμοποιούνται σε αυτό το κεφάλαιο. Πρώτον, θεωρούμε ένα ιστότοπο (website) ως έναν κατευθυνόμενο, μη κυκλικό γράφο, $G(V,E)$, όπου $|V| = n$, δηλαδή το website έχει n ιστοσελίδες. Έστω p_i η πιθανότητα πρόσβασης της ιστοσελίδας i . Όλα τα p_i διαμορφώνουν μια πιθανοτική κατανομή P . Η εντροπία του p ορίζεται ως

$$H(p) = -\sum_{i=1}^n p_i \log p_i . \text{ Σε κάθε ακμή } e(i,j) \text{ επίσης ανατίθεται ένα βάρος } s_{i,j} \in [0,1],$$

το οποίο συμβολίζει την ομοιότητα μεταξύ των ιστοσελίδων i και j . Για τους σκοπούς αυτής της εργασίας, οι εξερχόμενοι σύνδεσμοι της κάθε σελίδας i σχηματίζουν μια κατανομή, δηλαδή τα βάρη ομοιότητας κανονικοποιούνται σε κάθε κόμβο.

Το υπόλοιπο κεφάλαιο οργανώνεται ως εξής: Στη παράγραφο 3.2, παρέχεται μία εισαγωγή σε προηγούμενες σχετικές εργασίες. Στην επόμενη παράγραφο γίνεται η αρχιτεκτονική επισκόπηση του αλγόριθμου που λειτουργεί σαν οδηγός για το υπόλοιπο κεφάλαιο. Στη παράγραφο 3.4 εισάγεται ένα καινοτόμο μοντέλο και νέες μετρικές που μπορούν να χρησιμοποιηθούν για να μετρήσουμε την αποτελεσματικότητα και να αξιολογήσουμε την απόδοση του αλγόριθμου. Στη παράγραφο 3.5 περιγράφονται λεπτομερώς τα βήματα του προτεινόμενου αλγόριθμου. Στην παράγραφο 3.6 περιγράφεται το πειραματικό περιβάλλον, ενώ στην παράγραφο 3.7 παρουσιάζονται τα αποτελέσματα της εκτέλεσης του αλγόριθμου. Τέλος, στη παράγραφο 3.8 γίνεται η συνόψιση του κεφαλαίου και συζητούνται σχετικά μελλοντικά θέματα.

3.2 Προηγούμενες Εργασίες

Η έννοια των hotlinks εισήχθηκε από τους Perkowski και Etzioni [1], όπου ανάμεσα σε ένα σύνολο από άλλους χρήσιμους μετασχηματισμούς ιστοσελίδων, εισήγαγαν την έννοια του shortcutting, δηλαδή την πρόβλεψη κάποιων συνδέσμων. Μετά από αυτή τη μελέτη, ακολούθησε η δημοσίευση και άλλων σχετικών [2,3,4,5,6,7,8,9,10,11]. Αυτή η λίστα μελετών μπορεί να διαχωριστεί σύμφωνα με το συγκεκριμένο μοντέλο χρήστη. Ακολουθούν κυρίως δύο μοντέλα: το μαντικό μοντέλο χρήστη [2,3,8], και το μοντέλο άπληστου χρήστη [4,5,6,7,8,9,10,11]. Στο προηγούμενο μοντέλο, ο χρήστης έχει λάβει γνώση όλων των hotlinks που έχουν προστεθεί, και ως εκ τούτου και της δομής της ιστοσελίδας και καθίσταται για αυτόν δυνατό να ακολουθήσει ένα συντομότερο μονοπάτι από την αρχική σελίδα έως την επιθυμητή πληροφορία. Στο δεύτερο μοντέλο, σε κάθε κόμβο της δομής της νέας ιστοσελίδας, ο χρήστης έχει γνώση μόνο των hotlinks του παρόντος κόμβου και έτσι (εφόσον δεν έχει καθολική άποψη), παίρνει μία τοπική, άπληστη απόφαση, ακολουθώντας το link που τον οδηγεί πιο κοντά στην επιθυμητή πληροφορία. Οι Bose et al. [2] ήταν οι πρώτοι που μελέτησαν συστηματικά το πρόβλημα του μαντικού μοντέλου. Απέδειξαν ότι το πρόβλημα DAG είναι NP-hard (το ίδιο ισχύει και για το μοντέλο άπληστου χρήστη [6]) και επίσης εφάρμοσαν αλγόριθμους για την ανάθεση hotlinks, όπως επίσης και άνω και κάτω όρια για τον αναμενόμενο αριθμό βημάτων που απαιτούνται ώστε να φθάσουμε στα φύλλα από μία αρχική σελίδα, η οποία βρίσκεται στη ρίζα ενός πλήρους δυαδικού δέντρου. Όταν μία ιστοσελίδα λαμβάνεται υπόψη ως ένα δέντρο, οι Kranakis et al. [8] εφάρμοσαν έναν $O(n^2)$ αλγόριθμο, ο οποίος αναθέτει ένα hotlink σε κάθε κόμβο, έτσι ώστε το αναμενόμενο μήκος του μονοπατιού μετά από την ανάθεση να προσεγγίζει την εντροπία κατά μία σταθερά. Έχει ήδη αποδειχθεί ότι ο αλγόριθμος αυτός ξεπεράστηκε από άπληστες στρατηγικές (βλέπε τις ευρετικές μεθόδους που διατυπώνονται στο [3]).

Σε ό,τι αφορά το άπληστο μοντέλο, ένας βέλτιστος αλγόριθμος, του οποίου ο χρόνος εκτέλεσης και οι απαιτήσεις σε χώρο είναι εκθετικές σε σχέση με το βάθος του δέντρου (και έτσι πολυωνμικές για δέντρα λογαριθμικού βάθους) ανακαλύφθηκε ανεξάρτητα από τους Gerstel et al. [5] και από τους Pessoa et al. [10]. Στο [11], επιβεβαιώθηκε ότι η

υλοποίηση του αλγορίθμου αυτού είναι ικανή να βρίσκει βέλτιστες λύσεις για δέντρα τα οποία έχουν 10000 κόμβους και βάθος έως 14. Στο [7], αποδείχθηκε ότι η φυσική άπληστη στρατηγική επιτυγχάνει τουλάχιστον το μισό του ωφέλους μίας βέλτιστης λύσης στο μοντέλου άπληστου χρήστη. Μία άλλη προσέγγιση από την άποψη του κέρδους παρουσιάστηκε από τους Matichin and Peleg in [9]. Ο αλγόριθμός τους αναθέτει μόνο hotlinks, τα οποία προσπερνούν ακριβώς έναν κόμβο και εφαρμόζεται και στα δύο μοντέλα.

Οι Douieb and Langerman [12] παρουσίασαν ένας αλγόριθμο γραμμικού χρόνου στο παραπάνω πρόβλημα και κατάφεραν επίσης να διατηρήσουν αποτελεσματικά την ανάθεση των hotlinks με την παρουσία δυναμικών τελεστών (εισαγωγή κόμβων, διαγραφή, επανεκχώρηση βαρών). Ο αλγόριθμός τους χρησιμοποιεί τεχνικές από το [13]. Οι ίδιοι συγγραφείς [4] παρείχαν μία βελτίωση στον αλγόριθμο των Bose et al. [2], σημειώνοντας ότι η προσεγγιστική αναλογία του βέλτιστου μονοπατιού δεν συμπεριλαμβάνει τον παράγοντα d . Επινόησαν επίσης ένας αλγόριθμο ανάθεσης

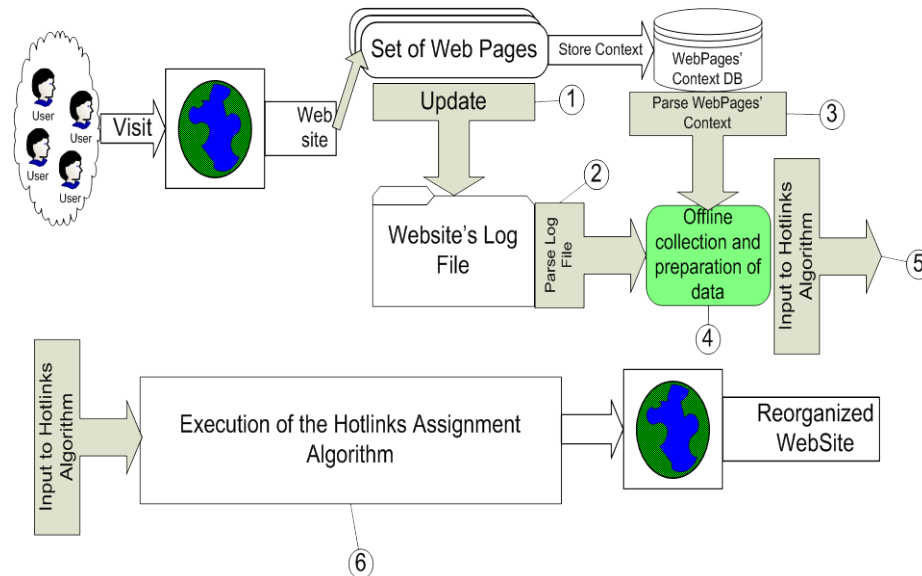
πολλαπλών hotlinks, με μέσο χρόνο προσπέλασης $\frac{H(P)}{(\log(k+d) - \log d)}$ και $O(n \log n)$ χρόνο εκτέλεσης.

Γενικά, έχει προταθεί ένας αξιοσημείωτος αριθμός προσεγγιστικών αλγορίθμων, όπως επίσης και όρια χειρότερης περίπτωσης για την ποιότητα των υπολογισμένων λύσεων. Παρ' όλα αυτά, λίγοι μόνο από αυτούς τους αλγορίθμους είναι γνωστοί για την πρακτική συμπεριφορά τους.

Στο [6], πραγματοποιήθηκε μία προσπάθεια εξαφάνισης αυτού του κενού, εκτιμώντας όλες τις πρόσφατες στρατηγικές πειραματικά. Τα πειράματα αυτά βασίζονται σε δέντρα τα οποία αποσπάστηκαν από πραγματικές ιστοσελίδες, όπως επίσης και σε συνθετικά δεδομένα.

Ένα κοινό στοιχείο όλων αυτών των μοντέλων είναι η υπόθεση ότι το συγκεκριμένο μοντέλο δέντρου (το οποίο φέρει το αρχικό περιεχόμενο των σελίδων) δε μεταβάλλεται και αποτελεί ένα δέντρο. Αντίθετα, το δικό μας μοντέλο μεταχειρίζεται τη δομή της ιστοσελίδας σαν κατευθυνόμενο άκυκλο γράφημα και λαμβάνει υπόψη το περιεχόμενο των σελίδων. Επιπρόσθετα, στο προτεινόμενο μοντέλο, στόχος δεν είναι η ελαχιστοποίηση μίας μετρικής στόχου, με βάση την προσβασιμότητα ολόκληρης της ιστοσελίδας, αλλά η βελτίωση τοπικά της κατάστασης κάθε σελίδας, βασιζόμενοι στην τοπική της προσβασιμότητα, η οποία εκφράζεται χρησιμοποιώντας μία κατάλληλη μετρική.

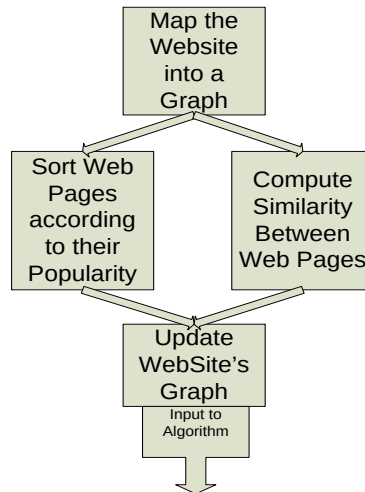
3.3 Αρχιτεκτονική Επισκόπηση



Εικόνα 17 Αρχιτεκτονική επισκόπηση της εφαρμογής του προτεινόμενου αλγόριθμου σε ένα website.

Στη συνέχεια, περιγράφεται η συνολική αρχιτεκτονική του σεναρίου της εφαρμογής του προτεινόμενου αλγόριθμου ανάθεσης hotlinks σε ένα website. Πρώτον, δεδομένου ενός website, οι χρήστες μπορεί να επισκεφτούν ένα σύνολο από ιστοσελίδες που ανήκουν σε αυτό. Κάθε επίσκεψη σε μια web σελίδα αυξάνει τη δημοτικότητα της και η επίσκεψη καταγράφεται στο logfile του website. Ανάλογα με το μέγεθος του website, μπορούμε να θέσουμε ένα κατώφλι επισκέψεων, πάνω από το οποίο μπορούμε να προτείνουμε ότι η εφαρμογή του αλγόριθμου θα βελτιώσει την οργάνωση και την λειτουργικότητα του website, όσον αφορά την ταχύτητα και την αποδοτικότητα της αναζήτησης της πληροφορίας που επιθυμεί ο χρήστης.

Σε αυτή την περίπτωση, δυο σημαντικά βήματα πρέπει να εκτελεστούν πριν την επεξεργασία των δεδομένων. Πρώτον πρέπει να αναλύσουμε το logfile του Website(βήμα 1). Αυτό το βήμα έχει ως σκοπό τη συγκέντρωση πληροφορίας που αφορά την τοπολογία και τη δημοτικότητα των web σελίδων μέχρι στιγμής. Δεύτερον, τα περιεχόμενα όλων των ιστοσελίδων πρέπει να αποθηκευθούν σε μια βάση δεδομένων(βήμα 2) έτσι ώστε να προσπελασθούν αργότερα (βήμα 3).



Εικόνα 18 Offline συλλογή και προετοιμασία των δεδομένων - Είσοδος στον αλγόριθμο ανάθεσης hotlinks.

Για την εισαγωγή δεδομένων στον αλγόριθμο (βήμα 5), απαιτείται αρχικά offline συλλογή των απαιτούμενων δεδομένων και στη συνέχεια προεπεξεργασία αυτών (Βήμα 4), όπως φαίνεται στην εικόνα 17. Πιο συγκεκριμένα, υπολογίζουμε την ομοιότητα περιεχομένου μεταξύ όλων των σελίδων ανά δύο και την αποθηκεύουμε σε έναν διδιάστατο πίνακα, $n \times n$, όπου n είναι ο αριθμός των ιστοσελίδων. Για να υπολογίζουμε την ομοιότητα μεταξύ σελίδων, έχουν προταθεί πολλοί αλγόριθμοι στο πεδίο της Ανάκτησης Πληροφορίας, [14][15]. Αφού ο διδιάστατος πίνακας ομοιότητας συμπληρωθεί, ταξινομούμε τις σελίδες ανάλογα με τη δημοτικότητα τους με βάση τις επισκέψεις χρηστών που έχουν καταγραφεί στο αντίστοιχο logfile του website.

Εισάγουμε τα συλλεγμένα και επεξεργασμένα δεδομένα στον αλγόριθμο (βήμα 5) και εκτελούμε τον αλγόριθμο ανάθεσης hotlinks (βήμα 6). Το αποτέλεσμα του αλγόριθμου θα είναι το ίδιο σύνολο ιστοσελίδων όπως πριν, αλλά οργανωμένο σε διαφορετική τοπολογία, αφού νέοι σύνδεσμοι θα έχουν τοποθετηθεί μεταξύ των ιστοσελίδων.

3.4 Μετρώντας την Απόδοση ενός Αλγόριθμου Ανάθεσης Hotlinks

3.4.1 Το Μοντέλο

Πριν την παρουσίαση του αλγορίθμου, θα οριστεί το μοντέλο και οι μετρικές για τη μέτρηση της απόδοσης του αλγορίθμου ανάθεσης hotlinks. Επομένως, περιγράφονται οι παράγοντες που λαμβάνονται υπόψη, έχοντας ως σκοπό να προταθεί ένα μοντέλο που θα συνυπολογίζει το περιεχόμενο των εμπλεκόμενων σελίδων.

Θα ακολουθήσει περιγραφή της προηγούμενης ακολοθούμενης προσέγγισης και πώς δύναται αυτή να τροποποιηθεί ώστε να ενσωματωθεί η ομοιότητα μεταξύ των σελίδων. Με βάση αυτή την προσέγγιση θα παρουσιαστεί η προτεινόμενη μετρική.

Το αρχικό βήμα είναι να ανατεθεί σε κάθε ακμή ένα βάρος που αντιστοιχεί στην ομοιότητα των δύο κόμβων που ενώνονται από αυτήν την ακμή. Το επόμενο βήμα είναι να μετασχηματίσουμε τη σχέση (1) έτσι ώστε να συμπεριλάβει τη μετρική ομοιότητας.

Μια πιθανή κατεύθυνση προς αυτό το στόχο, θα ήταν να εισαχθεί στην απόσταση μεταξύ του φύλλου i και της ρίζας. Το κίνητρο πίσω από αυτή τη συνένωση είναι ότι όσο μικρότερη είναι η ομοιότητα περιεχομένου μεταξύ των διατρεχόμενων κόμβων του μονοπατιού, τόσο μεγαλύτερη είναι η πραγματική απόσταση που πρέπει να διατρέξει ένας χρήστης για να πλησιάσει το μονοπάτι.

Εάν συμβολίσουμε με S_i το άθροισμα των ομοιοτήτων για κάθε ζευγάρι κόμβων που ενώνονται στο μονοπάτι από τον κόμβο προς τη ρίζα, η (1) μπορεί να μετασχηματιστεί ως εξής:

$$E[T^A, p] = \sum_{i_is_a_leaf} p_i d_A(i) / S_i$$

Μια διαφορετική προσέγγιση είναι να εφαρμόσουμε γινόμενα αντί για αθροίσματα. Σε αυτήν την περίπτωση, εάν P_i είναι το γινόμενο των ομοιοτήτων για κάθε ζευγάρι κόμβων που ενώνονται στο μονοπάτι από τον κόμβο προς τη ρίζα, η (1) μπορεί να μετασχηματιστεί ως εξής:

$$E[T^A, p] = \sum_{i_is_a_leaf} p_i d_A(i) / \prod_i$$

Και οι 2 προσεγγίσεις αξίζουν να ερευνηθούν αλλά ενδεχομένως να οδηγήσουν σε αλγορίθμους διαφορετικής διαίσθησης. Η προσέγγισή μας αποκλίνει από τη διαδικασία που μόλις περιγράφηκε, γιατί αντί να ελαχιστοποιούμε μια μετρική για όλο τον ιστότοπο, προτείνουμε μία μετρική που αφορά κάθε ιστοσελίδα του υπό επεξεργασία ιστότοπου και υποδηλώνει την προσβασιμότητα στην συγκεκριμένη σελίδα. Αυτή η μετρική επεκτείνει την Pagerank μετρική που εισάγεται στη μηχανή αναζήτησης Google.

Η αρχική Pagerank σχέση για μια web σελίδα A ορίζεται ως:

$$PR(A) = PR(T1)/L(T1) + \dots + PR(Tn)/L(Tn)$$

,όπου $PR(x)$ είναι το Pagerank του x , $L(x)$ είναι ο αριθμός των εξερχόμενων συνδέσμων σε μια σελίδα x . Επομένως για μια σελίδα u , αυτός ο τύπος συνοψίζεται ως εξής:

$$PR(u) = \sum_{u \in B_u} \frac{PR(u)}{L(u)}$$

Η Pagerank είναι μια μετρική για την αξιολόγηση του ανθρώπινου ενδιαφέροντος και προσοχής σε μια web σελίδα αλγοριθμικά και αντικειμενικά. Η αυθεντική Pagerank φόρμουλα για μια ιστοσελίδα ασχολείται με ολόκληρο το web, ενώ εδώ ασχολούμαστε με ιστοσελίδες που συνιστούν μόνο έναν ιστότοπο. Επομένως, στη συγκεκριμένη περίπτωση, θα χρησιμοποιηθεί μια παραπλήσια του Pagerank μετρική στην οποία θα αντανακλάται και η ομοιότητα περιεχομένου. Επομένως, αυτό το τροποποιημένο

Pagerank όπου λαμβάνεται υπόψιν και η ομοιότητα μεταξύ σελίδων θα ονομαστεί Simetric (Similarity Metric).

Πιο συγκεκριμένα, η Simetric υπολογίζεται ως εξής: Για σε κάθε ιστοσελίδα προσαρτείται μια ενδιάμεση τιμή που καλείται Aux που αρχικά έχει τιμή 0. Επιπλέον,

αρχικά η Simetric ισούται με $\frac{1}{\#webpages}$ και η τιμή μιας σελίδας υπολογίζεται

αναδρομικά χρησιμοποιώντας τις τιμές των ιστοσελίδων που δείχνουν σε αυτήν, γεγονός που σημαίνει πως οι ιστοσελίδες διανέμουν τη Simetric σε όλους τους εξερχόμενους συνδέσμους. Τυπικά, εάν για μια ιστοσελίδα w , συμβολίσουμε με $In(w)$ το σύνολο των ιστοσελίδων που έχουν συνδέσμους προς την w , τότε αποθηκεύουμε στη μεταβλητή Aux το άθροισμα της προηγούμενης Aux μεταβλητής της ιστοσελίδας και του γινομένου της Simetric τιμής όλων των ιστοσελίδων που ανήκουν στο $In(w)$ με την ομοιότητα των σελίδων με την w , διαιρεμένου με τον αριθμό των εξερχόμενων συνδέσμων. Η Aux(w) είναι αρχικά 0.

Έτσι:

$$Aux(w) = Aux(w) + \sum_{v \in In(w)} Simetric(v) * sim(v, w) / \#εξερχ \text{ _μενοι_σ _νδεσμοι_του_}$$

Επομένως, η Simetric της w υπολογίζεται ως εξής: $Simetric(w) = factor1 + (1 - factor2) * Aux(w)$, όπου οι σταθερές factor1 και factor2 είναι παράγοντες, οι τιμές των οποίων καθορίζονται από τον υλοποιητή. Συγκεκριμένα, ο παράγοντας factor1 ισούται με $q / \#webpages$ και ο factor2 ισούται με q , όπου q είναι ο παράγοντας απόσβεσης. Ένας υψηλός παράγοντας απόσβεσης σημαίνει πως τα αποτελέσματα είναι λιγότερα αποσβεσμένα και εφόσον υπάρχει μικρή απόσβεση, το Simrank που προέρχεται από εξωτερικούς συνδέσμους διαμοιράζεται στο σύστημα. Ο παραπάνω υπολογισμός επαναλαμβάνεται για όλες τις σελίδες του website, για έναν σταθερό αριθμό επαναλήψεων, έτσι ώστε να έχουμε την κατάλληλη μεταφορά βαρών από τη μία σελίδα στην άλλη, η Simetric να συγκλίνει και οι τιμές της Simetric που έχουμε λάβει για μέρος του γράφου να πλησιάζουν τις πραγματικές τιμές της Simetric (στα πειράματα που εκτελέσαμε ένας κατάλληλος αριθμός ήταν περίπου το εκατό). Ένα σαφές πλεονέκτημα του προτεινόμενου μοντέλου, σε σχέση με τις προηγούμενες προσεγγίσεις, είναι ότι δεν εξαρτάται από την υπόθεση της ύπαρξης μιας αρχικής σελίδας, που είναι η πηγή όλων των προσβάσεων στο site, αλλά υπολογίζει μια τιμή για κάθε σελίδα που δείχνει την προσβασιμότητά της ανάλογα με τους συνδέσμους που έχει η σελίδα στο συγκεκριμένο site και υποθέτοντας ότι κάθε σελίδα μπορεί να αποτελέσει την πηγή πρόσβασης στο website (αυτή η υπόθεση μοντελοποιεί αποδοτικότερα την πραγματικότητα, σε σύγκριση με την υπόθεση ότι η αρχική σελίδα είναι το μόνο σημείο εισόδου στο περιεχόμενο του website).

Βασιζόμενοι στην παραπάνω μετρική μπορούμε να ορίσουμε τον αλγοριθμικό μας στόχο ως εξής: δοσμένου ενός συνόλου ιστοσελίδων σε έναν ιστότοπο, ταξινομημένων σύμφωνα με τη δημοτικότητά τους, προσθέτουμε hotlinks (ένα ανά σελίδα), με σκοπό η κατάταξη των σελίδων σύμφωνα με το Simetric να είναι όσο το δυνατόν πλησιέστερα στην κατάταξή τους σύμφωνα με τη δημοτικότητά τους. Συνεπώς, η κεντρική ιδέα της Simetric είναι διπλή. Σκοπός είναι να συνδεθούν ιστοσελίδες με σημασιολογικά κοντά σε αυτές άλλες ιστοσελίδες ανάλογα με τη δημοτικότητά τους. Για την επίτευξη αυτών των

δύο παραμέτρων, λαμβάνονται υπόψιν (α) η δημοτικότητα της ιστοσελίδας και (β) η ομοιότητα της δεδομένης ιστοσελίδας με τις υπόλοιπες ιστοσελίδες του ιστότοπου. Η δημοτικότητα της ιστοσελίδας χρησιμοποιείται για να προδιαθέτει και να ορίζει τον αριθμό νέων συνδέσμων με ανάλογο τρόπο, αλλά ταυτόχρονα ελέγχεται μέσω της ομοιότητας μεταξύ της τρέχουσας ιστοσελίδας και της ιστοσελίδας με την οποία πρόκειται να συνδεθεί. Σαν αποτέλεσμα, η επίτευξη μιας κατάταξης με βάση τη Simetric με τιμές κοντά στην κατάταξη σύμφωνα με τη δημοτικότητα, σημαίνει ότι όσο πιο δημοφιλείς είναι οι σελίδες του γράφου που προκύπτει, με τόσες περισσότερες σχετικές σελίδες συνδέονται.

3.4.2 Οι μέθοδοι αξιολόγησης

Με σκοπό να μετρηθεί η απόδοση του αλγόριθμου, προτείνονται τέσσερις μετρικές που μπορούν αποδοτικά να καταδείξουν τον τρόπο με τον οποίο ο αλγόριθμος επηρεάζει την τοπολογία του website και πιθανώς την λειτουργικότητα από διαφορετικές οπτικές γωνίες. Ο στόχος είναι να χρησιμοποιηθούν τέτοιες μετρικές που να είναι σε θέση να αποδείξουν ότι η νέα τοπολογία του ιστότοπου πράγματι υπηρετεί τους σκοπούς για τους οποίους ο αλγόριθμος σχεδιάστηκε.

3.4.2.1 Μετρική απόστασης

Η πρώτη μετρική συγκρίνει την κατάταξη των σελίδων με βάση τη Simetric σε σχέση με την κατάταξη τους ως προς τη δημοτικότητά τους. Ένας τρόπος για να γίνει αυτό είναι να συγκριθούν οι δύο ακολουθίες ανά ζευγάρια. Πιο συγκεκριμένα, συγκρίνουμε κάθε θέση στη λίστα των ταξινομημένων σελίδων με βάση τη δημοτικότητά τους, με την ίδια θέση στη λίστα με τις σελίδες ταξινομημένες με βάση τη Simetric. Έτσι, η μετρική μας είναι το πηλίκο του αριθμού των ταιριασμένων ζευγαριών, διαιρεμένου με n . Έστω μια κατάταξη των σελίδων με βάση τη δημοτικότητά τους και με $gr(i)$ συμβολίζεται η κατάταξη της σελίδας i . Παρόμοια, έστω η κατάταξη των σελίδων με βάση την προσβασιμότητά τους και με $rs(i)$ συμβολίζεται η ανάλογη κατάταξη της σελίδας, όπως αυτή προκύπτει από τη Simetric. Η ομοιότητα μεταξύ των δύο κατατάξεων υπολογίζεται μέσω του τύπου:

$$\sum_i |r_s(p_i) - r_p(p_i)|$$

Όσο πιο κοντά στο μηδέν είναι το αποτέλεσμα, τόσο πιο αποτελεσματικός θεωρείται ο αλγόριθμος.

3.4.2.2 Μετρική Βάρους

Και σε αυτήν την περίπτωση, συμβολίζουμε με $gr(i)$ και $rs(i)$ την κατάταξη της σελίδας i ως προς τη δημοτικότητά της και τη Simetric αντίστοιχα. Έστω $\max(i) = \max\{gr(i), rs(i)\}$ και $\min(i) = \min\{gr(i), rs(i)\}$. Τότε, η ομοιότητα μεταξύ των δύο κατατάξεων υπολογίζεται μέσω του τύπου:

$$n - \sum_{i=1}^n \frac{\min(p_i)}{\max(p_i)}.$$

Όσο πιο κοντά στο 0 βρίσκεται ο υπολογισμός αυτής της σχέσης, τόσο πιο αποτελεσματικός μπορεί να θεωρηθεί ο αλγόριθμος. Η μετρική βάρους ακολουθεί ακριβώς την ίδια λογική με τη μετρική απόστασης και στα πειράματα ο υπολογισμός ενός εκ των δύο αρκεί.

3.4.2.3 Μετρική με Βάση την Εντροπία

Ένας άλλος τρόπος για τη μέτρηση της απόδοσης του αλγορίθμου, δεδομένου ενός ιστότοπου, είναι να μετρήσουμε την εντροπία του ιστότοπου πριν και μετά την εφαρμογή του αλγορίθμου. Αναπαριστούμε κάθε σελίδα με μια κατάσταση και αναθέτουμε πιθανότητες στους συνδέσμους, καταλήγοντας σε μια αλυσίδα Markov. Αυτές οι πιθανότητες είναι στην πραγματικότητα οι ομοιότητες μεταξύ των σελίδων. Στο μοντέλο που περιγράφεται στο [16], κάθε σύνοδος πλοήγησης του χρήστη μπορεί να θεωρηθεί ως το ίχνος μιας εργοδικής αλυσίδας Markov, έτσι ώστε όταν μία σύνοδος ολοκληρωθεί, μία νέα να μπορεί να αρχίσει σε οποιαδήποτε κατάσταση με μη μηδενική πιθανότητα σύμφωνα με μια εμπειρική αρχική κατανομή. Μια εργοδική Markov αλυσίδα είναι μια πεπερασμένη Markov αλυσίδα μη περιοδική και μη μειώσιμη. Πιο συγκεκριμένα, $G=(N;E)$ είναι ο αντίστοιχος πεπερασμένος κατευθυνόμενος γράφος. Οι κόμβοι στο N καλούνται καταστάσεις, η πληθικότητα του N είναι n , και η πληθικότητα του E καλείται μέγεθος του G . Επιπρόσθετα, οι πιθανότητες P_{ij} που ανατίθενται στους συνδέσμους καλούνται πιθανότητες μετάβασης. Σημειώνουμε ότι αφού η Markov αλυσίδα είναι εργοδική, για κάποιο $m>0$, για όλα τα i,j $P_{mij} > 0$. Η εντροπία μιας Markov αλυσίδας δίνεται από την σχέση

$$H(M) = - \sum_{i=1}^n \sum_{j=1}^n \pi_i P_{ij} \log P_{ij}$$

,όπου το π υποδηλώνει την επιτυγχανόμενη στατική κατανομή. Να σημειωθεί ότι στο [29] παρουσιάζεται μια καινοτόμος μέθοδος για online αυξητικό υπολογισμό της εντροπίας, εξομοιώνοντας μια τυχαία πλοήγηση ενός χρήστη σε ένα website.

3.4.2.4 Μετρική με Βάση το Μήκος Μονοπατιού

Η τελευταία επιλογή είναι να μετρηθούν τα μήκη των μονοπατιών και να ελεγχθεί εάν αυτά ελαττώνονται. Εάν το μέσο μήκος μονοπατιού από την αρχική σελίδα σε μια σελίδα ελαττωθεί, αυτό θα σήμαινε ότι γενικά ένας χρήστης θα μπορούσε να έχει πρόσβαση σε πληροφορίες γρηγορότερα και έτσι η προσβασιμότητα στο website θα βελτιωνόταν. Για τον υπολογισμό αυτής της μετρικής χρειάζεται μόνο να εντοπιστούν τα συντομότερα μονοπάτια από ένα συγκεκριμένο κόμβο-πηγή (η αρχική σελίδα στην περίπτωση μας) προς κάθε άλλο κόμβο στον κατευθυνόμενο, με βάρη γράφο. Και σε αυτήν την περίπτωση, αυτή η μετρική είναι στη λογική της προηγούμενης προσέγγισης, αφού υποθέτει ότι η αρχική σελίδα είναι η βασική πηγή προσβάσεων στο website, αλλά θα ήταν ενδιαφέρον να αξιολογηθεί η αποδοτικότητα του αλγορίθμου μετρώντας την.

Αφού ορίστηκε ένα γενικευμένο μοντέλο αλλά και κατάλληλες μετρικές για την αξιολόγηση ενός αλγορίθμου ανάθεσης hotlink, στην επόμενη παράγραφο παρουσιάζεται αναλυτικά η αλγοριθμική προσέγγιση.

3.5 Ο Αλγόριθμος

Σε αυτήν την παράγραφο παρουσιάζεται ένας τυχαιοποιημένος αλγόριθμος, ο οποίος λαμβάνει υπόψιν την δημοτικότητα των ιστοσελίδων και την ομοιότητα μεταξύ τους και προτείνει την ανάθεση hotlinks, ενός hotlink ανά σελίδα. Ο αλγόριθμος αποτελείται από 4 βασικά βήματα:

Βήμα1. Το αρχικό σύνολο P όλων των ιστοσελίδων χωρίζεται σε δύο υποσύνολα, που ονομάζονται POP και NonPOP (popular, non-popular). Στην περίπτωση των n ιστοσελίδων, το σύνολο P θα περιλαμβάνει τις $n/2$ πιο δημοφιλείς σελίδες και το σύνολο NonPOP τις $n/2$ λιγότερο δημοφιλείς σελίδες. Κάθε σελίδα έχει ως ιδιότητα τη δημοτικότητά της. Όσον αφορά τις ακμές, έχουν ως βάρος την ομοιότητα μεταξύ των κόμβων τους οποίους συνδέουν.

Βήμα2 Διαλέγουμε έναν κόμβο από κάθε σύνολο. Ο κόμβος από το σύνολο POP επιλέγεται σύμφωνα με την πιθανοτική κατανομή, η οποία ορίζεται σε σχέση με τη δημοτικότητά του. Για παράδειγμα, εάν ο κόμβος i έχει δημοτικότητα p , τότε επιλέγεται με πιθανότητα $p/(\text{Άθροισμα όλων των δημοτικοτήτων})$. Όσον αφορά τη δημοτικότητα, μπορούμε να εκφραστεί ως συχνότητα προσβάσεων ανά συνδέσμους που ακολουθούνται. Με άλλα λόγια, η δημοτικότητα κάθε σελίδας μετρείται ως ο αριθμός επισκέψεων από χρήστες που έχουν διανύσει ένα μονοπάτι από k συνδέσμους για να φτάσουν τη συγκεκριμένη σελίδα. Αυτά τα μονοπάτια μπορούν να καταλήξουν σε αυτή τη σελίδα ή να περάσουν από αυτήν.

Η επόμενη πράξη είναι να επιλεγεί ένας κόμβος από το δεύτερο σύνολο. Έστω ότι επιλέχθηκε ο κόμβος i από το σύνολο POP. Επιλέγονται οι κόμβοι που αντιστοιχούν στο μονοπάτι με τη μεγαλύτερη συχνότητα από την πηγή προς τον i . Για να υπολογιστεί η δημοτικότητα των μονοπατιών, επιλέγεται αυθαίρετα μια σελίδα, η οποία ορίζεται ως η αρχική σελίδα του site. Στη συνέχεια, υπολογίζονται όλα τα μονοπάτια που οδηγούν στον i . Όσον αφορά την δημοτικότητα κάθε μονοπατιού, για κάθε κόμβο διαιρείται η δημοτικότητά του με τον αριθμό εισερχόμενων ακμών και τα αποτελέσματα προστίθενται για όλους τους κόμβους του μονοπατιού. Αφού έχει επιλεγεί το πιο δημοφιλές μονοπάτι που οδηγεί στον κόμβο i , P , επιλέγονται οι κόμβοι j , έτσι ώστε ο κοντινότερος κοινός πρόγονός τους να ανήκει στο πιο δημοφιλές μονοπάτι που οδηγεί στον κόμβο i και η πρόσθεση της ακμής (i,j) καταλήγει σε ένα συντομότερο μονοπάτι για το i , από ό,τι το αρχικό πιο δημοφιλές. Ταξινομούμε τους υποψήφιους κόμβους σύμφωνα με την ομοιότητα με τον κόμβο i .

Βήμα3. Συνδέουμε μέσω ενός νέου συνδέσμου, τον κόμβο i με τον κόμβο που είναι πιο όμοιος και διαγράφουμε τον δεύτερο από το σύνολο NonPOP. Πρέπει να αναθέσουμε ένα βάρος στο νέο σύνδεσμο. Μπορούμε να πάρουμε αυτή την πληροφορία από τον αντίστοιχο πίνακα. Μετά, πρέπει να κανονικοποιήσουμε τα βάρη των εξερχόμενων συνδέσμων του δεύτερου κόμβου.

Βήμα4. Η διαδικασία επαναλαμβάνεται μέχρι να αδειάσει το NonPOP. Στη συνέχεια δημιουργούνται δύο νέα σύνολα POP και NonPOP από το παλιό POP και εκτελείται ο αλγόριθμος αναδρομικά.

Παρακάτω παρουσιάζεται ο ψευδοκώδικας του αλγόριθμου:

1. **HotlinkAssign(G)**
2. **If (G not singular) {**
3. **Initialize:**
4. Sort nodes according to popularity.

```

5. Assign nodes to sets POP and NonPOP.
6. While (NonPOP not empty) {
7.     Choose node  $i$  from set POP.
8.     In order to choose nodes from set NonPOP.
9.         Locate the most popular path  $P$  to  $i$  from the homepage (a
            randomly selected page).
10.        Choose nodes  $j$  such that their nearest common ancestor with
             $i$  belongs to  $P$  and the addition of the edge  $(j,i)$  results
            to a shortest path to  $i$ , than the initial most popular.
11.        Sort the candidate nodes according to their similarity with  $i$ .
12.        Choose the most similar with  $i$  node,  $y$ .
13.    Connect via hotlink  $(y,i)$  nodes  $i$  and  $y$ .
14.    Delete node  $y$  from NonPOP.
15.    Assign weight to the new link.
16.    Normalize weights of outgoing links of  $y$ .
17. } //end of While
18. HotlinkAssign( $P$ )
19. end;
    
```

Ο προτεινόμενος αλγόριθμος βασίζεται σε τρεις βασικές αλγοριθμικές συνιστώσες:

1. Έναν αλγόριθμο ταξινόμησης. Ο καλύτερος αλγόριθμος ταξινόμησης για n στοιχεία εισόδου απαιτεί χρόνο $O(n \log n)$ και γραμμικό χώρο στην χειρότερη περίπτωση [17]. Η ανάθεση των κόμβων στα σύνολα POP και NonPOP, μπορεί να εκτελεστεί σε $O(1)$ χρόνο για κάθε κόμβο.

2. Έναν αλγόριθμο εύρεσης ελάχιστου μονοπατιού. Για τους σκοπούς των πειραμάτων χρησιμοποιείται ο Bellman Ford αλγόριθμος [18]. Η χρονική πολυπλοκότητα του Bellman Ford αλγόριθμου είναι $O(n \cdot m)$ για έναν γράφο με n κόμβους και m ακμές.

3. Έναν αλγόριθμο NCA(Κοντινότερος Κοινός Πρόγονος) για κατευθυνόμενα άκυκλα γραφήματα (DAGs). Το καλύτερο γνωστό αποτέλεσμα για NCA σε DAGs παρουσιάζεται στο [19], όπου η πρώτη μέθοδος που προτείνεται είναι αρκετά απλή και λύνει το NCA πρόβλημα για τον DAG σε n φύλλα και m ακμές σε σταθερό χρόνο μετά από $O(nm)$ επανεπεξεργασία.

3.6 Πειράματα

Με σκοπό την μέτρηση της απόδοσης του αλγορίθμου, εκτελέστηκαν πειράματα χρησιμοποιώντας σαν είσοδο συνθετικά άλλα και πραγματικά δεδομένα. Αναπτύχθηκε ένας προσομοιωτής website σε C++, χρησιμοποιώντας τη LEDA, τη βιβλιοθήκη τύπων δεδομένων και αλγορίθμων [20].

Ο στόχος είναι η αναπαράσταση ενός website μέσω ενός γράφου και εκτελώντας τον αλγόριθμο πάνω στον γράφο, να εκτιμηθούν τα αποτελέσματά του. Για να εκτελεστεί ένα πείραμα ακολουθείται η παρακάτω διαδικασία: Αρχικά, δημιουργείται ένας συνεκτικός κατευθυνόμενος γράφος. Επιλέγονται ο αριθμός των κόμβων που αναπαριστούν τις ιστοσελίδες και ο αριθμός των ακμών που θα τοποθετηθούν στον γράφο, συμβολίζοντας τους συνδέσμους μεταξύ των ιστοσελίδων. Τέλος, για τη δημιουργία του κατευθυνόμενου άκυκλου γράφου, διαγράφονται όλοι οι κύκλοι που ανακαλύπτονται στο γράφο, διαγράφοντας τις αντίστοιχες ακμές. Μετά την κατασκευή του γράφου, επιλέγεται τυχαία η αρχική σελίδα του γράφου. Επίσης ανατίθενται τυχαία βάρη σε όλους τους κόμβους και σε όλες τις ακμές, που αναπαριστούν τη δημοτικότητα των κόμβων και την ομοιότητα μεταξύ τους, αντίστοιχα.

Για να υπολογιστεί το πιο δημοφιλές μονοπάτι από την αρχική σελίδα σε κάθε κόμβο /ιστοσελίδα, χρησιμοποιείται ο Bellman-Ford αλγόριθμος [18], ο οποίος υπολογίζει τα συντομότερα μονοπάτια σε έναν γράφο με βάρη και μοναδική πηγή (όπου κάποια από τα βάρη των ακμών μπορεί να είναι αρνητικά).

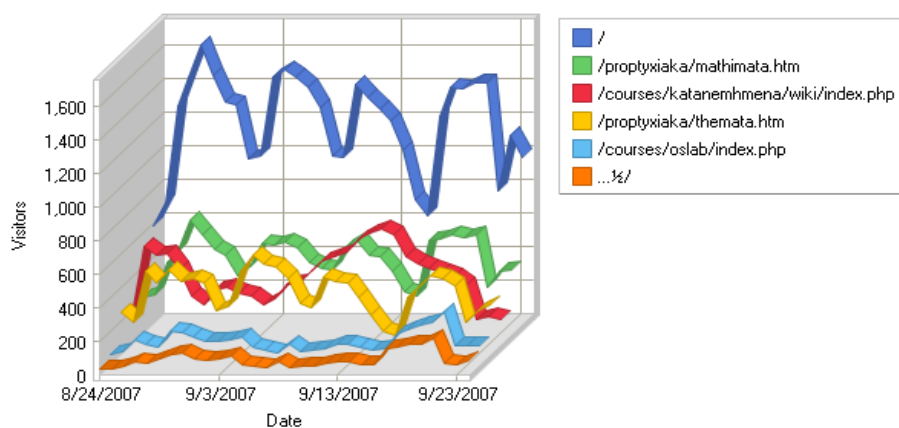
Τα πλεονεκτήματα της χρήσης του προσομοιωτή είναι ότι χρησιμοποιώντας τον ίδιο αριθμό κόμβων και ακμών σε κάθε πείραμα, εκτελείται ο αλγόριθμος σε διαφορετικές τοπολογίες κάθε φορά με σκοπό τη μελέτη των διακυμάνσεων της απόδοσής του, και επίσης επιλέγοντας διαφορετικούς αριθμούς ακμών και κόμβων, διαφαίνονται οι διαφορές στην απόδοση του αλγορίθμου για ένα μεγάλο website σε σχέση με τα αποτελέσματα σε ένα μικρό website.

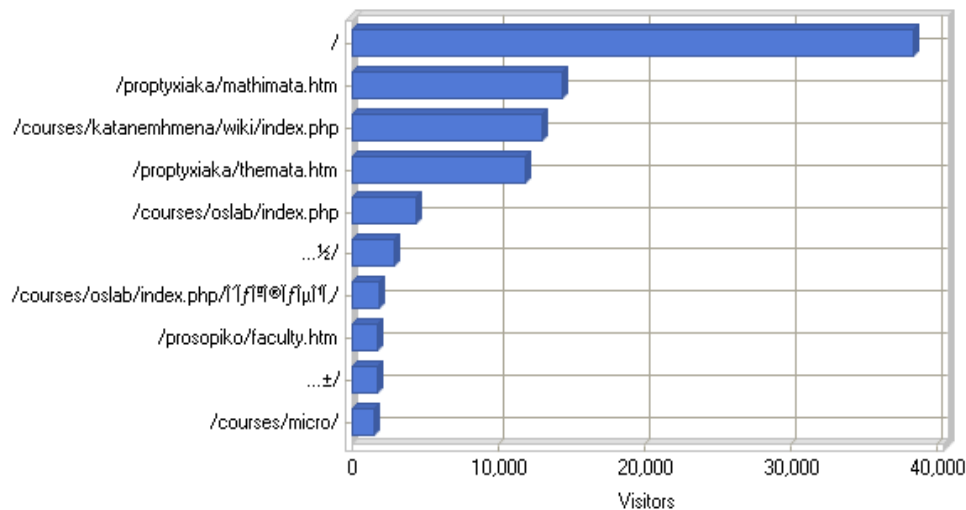
Από την άλλη πλευρά, θα είχε μεγάλη αξία να ελεγχθεί η απόδοση του αλγορίθμου σε ένα πραγματικό website. Πράγματι, ο αλγόριθμος εφαρμόστηκε στο site του τμήματος Μηχανικών Η/Υ κ Πληροφορικής του Πανεπιστημίου Πατρών, www.ceid.upatras.gr. Η δομή του site όμως είναι τέτοια που δεν βοηθά στην κατάδειξη των δυνατοτήτων του αλγορίθμου, ο οποίος αποδίδει πολύ καλύτερα για ιστότοπους με μεγάλο βάθος. Πιο συγκεκριμένα, όπως φαίνεται στα δύο επόμενα σχήματα, το site έχει ένα μέσο μέγιστο βάθος των τριών σελίδων και η πλειοψηφία των δημοφιλών σελίδων βρίσκεται σε βάθος δύο. Έτσι, δεν υπάρχει λόγος να ανατεθούν hotlinks σε αυτές, αφού η απόστασή τους από την αρχική σελίδα είναι μόνο ένας σύνδεσμος. Παρόλα αυτά, πολύ λίγα hotlinks ανατίθενται, αλλά η επίδρασή τους στο τελικό αποτέλεσμα δεν είναι σημαντική. Όμως, ακόμη και κάτω από αυτές τις συνθήκες, τα αποτελέσματα είναι πολύ θετικά αφού στην περίπτωση των δημοφιλών σελίδων με βάθος 3, ανατέθηκαν σε αυτές hotlinks κατευθείαν από την αρχική σελίδα, μειώνοντάς το βάθος τους κατά ένα. Για να δοκιμαστεί ο αλγόριθμος στο site, αναπαραστάθηκε το site σε έναν γράφο και αναλύθηκε με τη βοήθεια του προσομοιωτή. Όσον αφορά τον υπολογισμό της ομοιότητας περιεχομένου μεταξύ των σελίδων, χρησιμοποιήθηκε ένα online εργαλείο, το Similar Page Checker[21]. Το εργαλείο αυτό επιτρέπει σε κάποιον να υπολογίσει το ποσοστό ομοιότητας μεταξύ δυο σελίδων. Αυτό που παρατηρείται, είναι ότι σελίδες όπως οι δημοφιλείς download σύνδεσμοι, δεν συνδέονται απευθείας με την αρχική σελίδα, αφού η ομοιότητα περιεχομένου μεταξύ τους είναι πολύ μικρή. Παρακάτω παρουσιάζεται η περίληψη του logfile του site.

Hits	
Total Hits	1,755,559

Average Hits per Day	54,861
Average Hits per Visitor	17.47
Cached Requests	595,908
Failed Requests	22,819
Page Views	
Total Page Views	290,394
Average Page Views per Day	9,074
Average Page Views per Visitor	2.89
Visitors	
Total Visitors	100,512
Average Visitors per Day	3,141
Total Unique Ips	26,959
Bandwidth	
Total Bandwidth	22.43 GB
Average Bandwidth per Day	717.64 MB
Average Bandwidth per Hit	13.39 KB
Average Bandwidth per Visitor	233.96 KB

Πίνακας 4 Περίληψη του logfile του www.ceid.upatras.gr



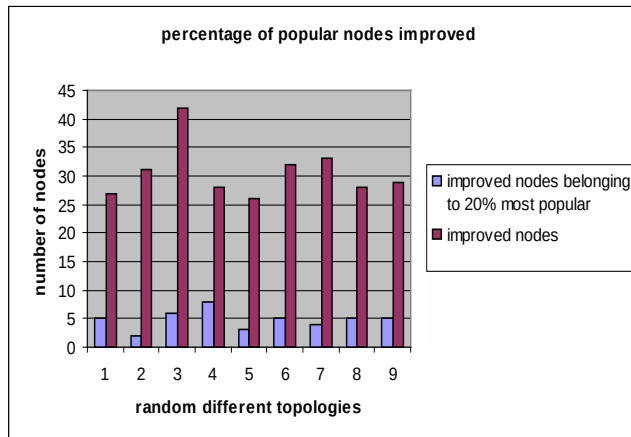


Εικόνα 19 Στατιστικά δεδομένα επισκεψιμότητας του www.ceid.upatras.gr

3.7 Αξιολόγηση

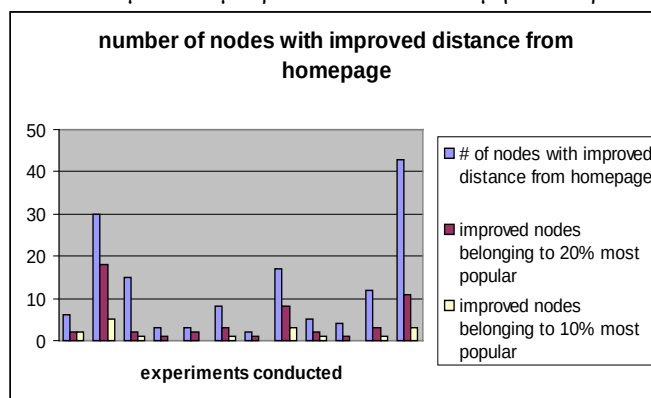
Στα πειράματα που περιγράφονται παρακάτω, χρησιμοποιήθηκαν συνεκτικοί γράφοι 100 και 500 κόμβων. Έχουν διεξαχθεί πολλά πειράματα χρησιμοποιώντας τον ίδιο αριθμό κόμβων, αλλά με διαφορετική τοπολογία κάθε φορά. Τα αποτελέσματα που φαίνονται στις γραφικές είναι αντιπροσωπευτικά της μέσης τιμής των αποτελεσμάτων μετά από την εκτέλεση του αλγορίθμου 100 φορές σε 100 διαφορετικές τοπολογίες. Ο κύριος σκοπός του αλγορίθμου είναι η διαμόρφωση της τοπολογίας του ιστότοπου, ώστε να προωθείται το δημοφιλές περιεχόμενο. Στην γραφική παράσταση 20 παρουσιάζεται το ποσοστό των κόμβων των οποίων η Simetric έχει βελτιωθεί και αντιπροσωπεύουν το 10%, 20% και 50% των πιο δημοφιλών κόμβων αντίστοιχα, σε ένα σύνολο 500 κόμβων για διαφορετική τοπολογία κάθε φορά. Στην περίπτωση των 100 κόμβων παρατηρούμε ότι το ποσοστό των κόμβων που βελτιώθηκαν όσον αφορά τη Simetric είναι παρόμοιο με την περίπτωση των 500 κόμβων. Η διαφορά με την περίπτωση των 500 κόμβων είναι ότι το ποσοστό των δημοφιλών κόμβων που έχουν βελτιωθεί στην περίπτωση των 100 κόμβων είναι μικρότερο. Το συμπέρασμα που προκύπτει είναι ότι ο αλγόριθμος αποδίδει καλύτερα για μεγάλους ιστότοπους από ότι για μικρούς. Αυτό το γεγονός δεν είναι μόνο φυσιολογικό και αναμενόμενο αλλά και επιθυμητό ταυτόχρονα. Συγκεκριμένα σε ιστότοπους που αποτελούνται από πολλές σελίδες, οι δημοφιλείς σελίδες συμμετέχουν με μεγάλη πιθανότητα σε πολλές αναθέσεις hotlinks και όσο περισσότερες ιστοσελίδες υπάρχουν τόσες περισσότερες αναθέσεις γίνονται και οι δημοφιλείς σελίδες ευνοούνται εφόσον επιλέγονται να δεχθούν hotlink με πιθανοτική κατανομή ανάλογη της δημοτικότητάς τους. Από την άλλη πλευρά σε μικρούς ιστότοπους δεν υπάρχει η ανάγκη να βελτιωθεί η προσβασιμότητα στις δημοφιλείς σελίδες σε σχέση με την απόσταση από την αρχική σελίδα, λόγω της ίδιας τους της φύσης. Με άλλα λόγια, σε μικρούς

ιστότοπους αναμένεται ότι η απόσταση από τη βασική σελίδα στις δημοφιλείς σελίδες θα είναι πολύ μικρή και για αυτό η πρόσβαση σε αυτές θα είναι έτσι και αλλιώς εύκολη.

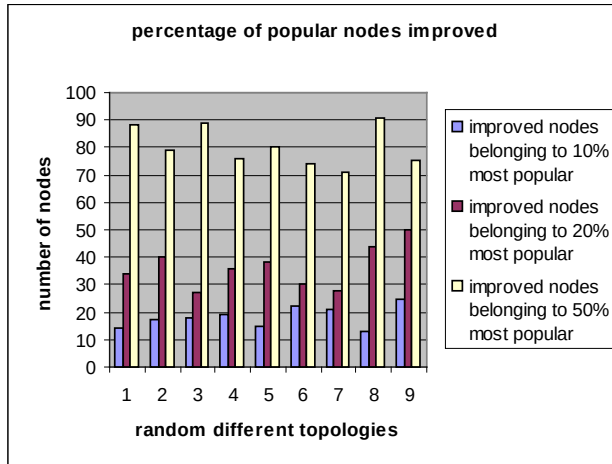


Εικόνα 20 Ποσοστό κόμβων που βελτιώσαν τη Simetric μετά από την εφαρμογή του αλγορίθμου σε σύνολο 100 κόμβων

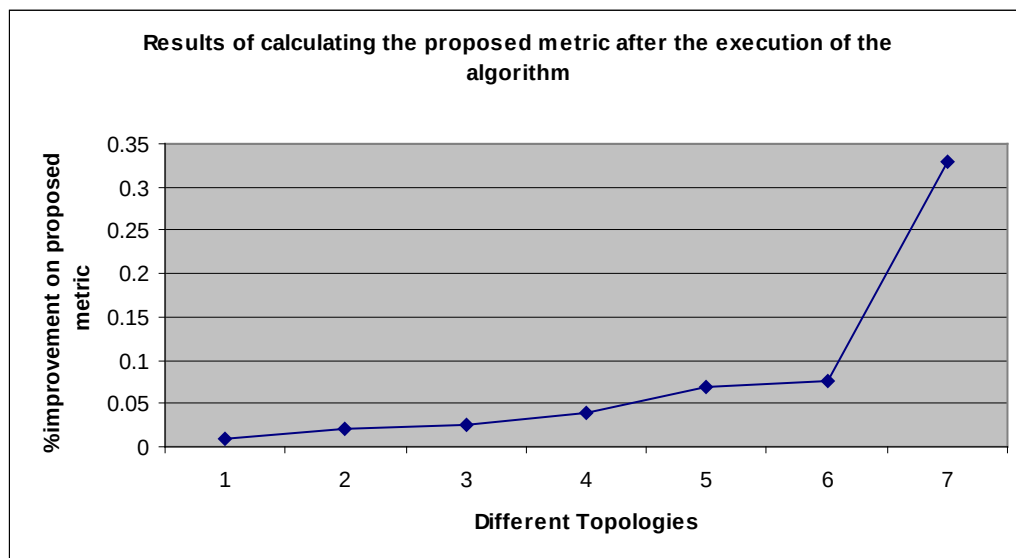
Η γραφική παράσταση 21 δείχνει τις βελτιώσεις που επιτυγχάνονται στα μήκη μονοπατιών. Υπολογίζονται τα συντομότερα μονοπάτια από την αρχική ιστοσελίδα προς κάθε κόμβο πριν και μετά την εκτέλεση του αλγορίθμου χρησιμοποιώντας τον αλγόριθμο Bellman-Ford. Ανάλογα με τη τοπολογία του ιστότοπου, ο αριθμός των κόμβων των οποίων τα μονοπάτια έχουν μειωθεί ανέρχεται στο 10% του αριθμού των συνολικών κόμβων. Η γραφική παράσταση 21 δείχνει ότι όσον αφορά την απόσταση από την αρχική σελίδα, το ποσοστό των κόμβων που βελτιώνεται είναι από 1% ως 10%. Λαμβάνοντας υπόψιν ότι ο γράφος είναι ισχυρά συνεκτικός, τα αποτελέσματα είναι πολύ ενθαρρυντικά. Για την ακρίβεια, οι γράφοι που έχουν δημιουργηθεί είναι πολύ πυκνοί και ένας χρήστης μπορεί να ακολουθήσει πολλαπλά μονοπάτια για να φτάσει σε έναν κόμβο. Επομένως, εφόσον επιτυγχάνεται μείωση απόστασης κατά 10% στους κόμβους, αυτό σημαίνει ότι σε πραγματικά site όπου οι τοπολογίες έχουν έλλειψη συνεκτικότητας, τα αποτελέσματα θα μπορούσαν να είναι ακόμη καλύτερα.



Εικόνα 21 Βελτιώσεις που επιτυγχάνονται στα μήκη μονοπατιών



Εικόνα 22 Ποσοστό κόμβων που βελτιώσαν τη Simetric μετά από την εφαρμογή του αλγορίθμου σε σύνολο 100 κόμβων

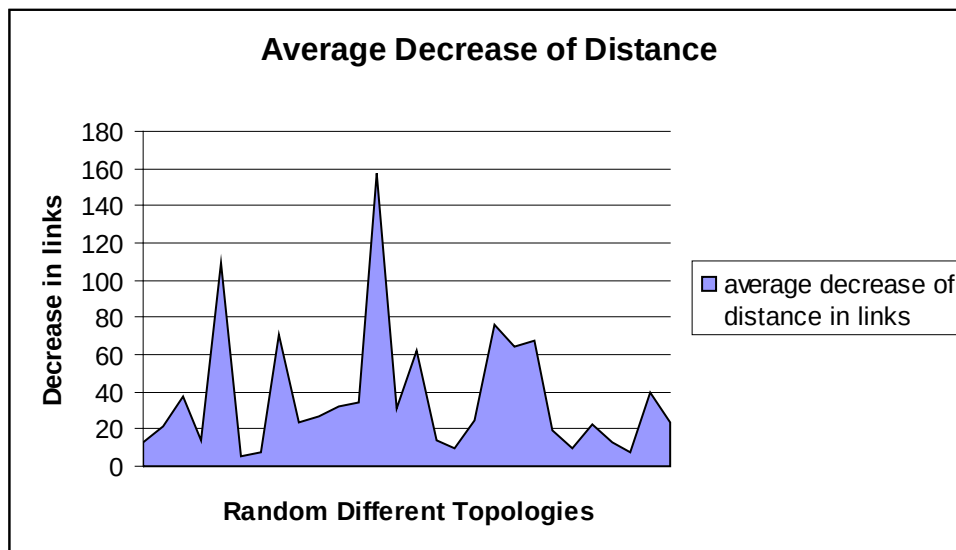


Εικόνα 23 Ποσοστό της βελτίωσης σε σχέση με τη μετρική απόστασης.

Όπως αναφέρθηκε νωρίτερα, στόχος είναι η σύγκριση της κατάταξης των σελίδων σύμφωνα με τη Simetric με την κατάταξή τους με βάση τη δημοτικότητά τους πριν και μετά την εκτέλεση του αλγορίθμου. Η γραφική παράσταση 23 δείχνει το ποσοστό της βελτίωσης σε σχέση με την πρώτη μετρική, τη μετρική απόστασης.

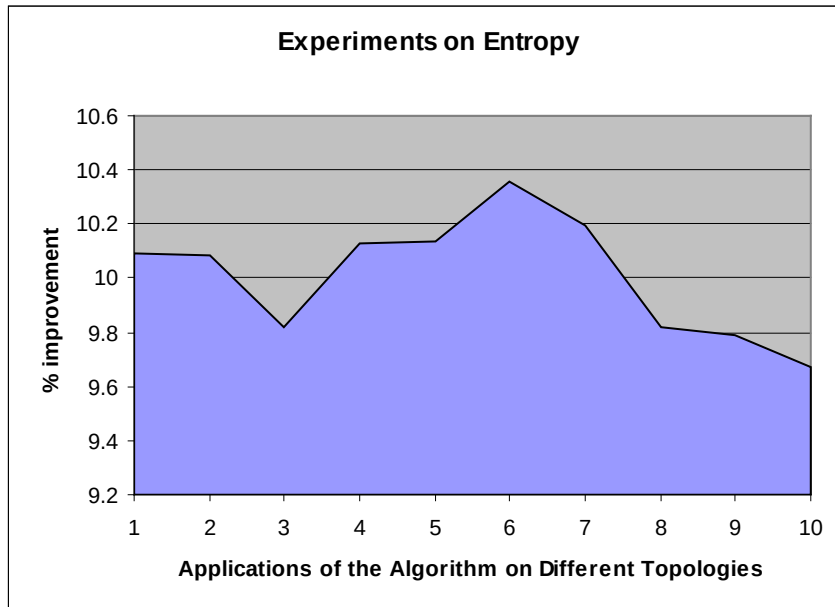
Όπως βλέπουμε στη γραφική παράσταση 23 η βελτίωση της μετρικής απόστασης φτάνει το 34%. Αυτό σημαίνει ότι μετά την εκτέλεση του αλγορίθμου το άθροισμα αποστάσεων μειώθηκε κατά 34%. Όπως έχει ήδη αναφερθεί, η βέλτιστη περίπτωση είναι

όταν αυτό το άθροισμα ισούται με μηδέν. Σε αυτή τη περίπτωση, το 34% των κόμβων μετά την εκτέλεση του αλγορίθμου έχει κατάταξη με βάση τη Simetric ίδια με την κατάταξη με βάση τη δημοτικότητα. Αυτό σημαίνει ότι πλέον περισσότεροι δημοφιλείς κόμβοι έχουν μεγαλύτερη τιμή Simetric και επομένως η πρόσβαση σε αυτούς θα προωθείται στο μέλλον. Τα αποτελέσματα της εφαρμογής του αλγορίθμου σε σχέση με τη δεύτερη μετρική, τη μετρική βάρους, είναι σχεδόν ίδια με αυτά που μόλις αναφέρθηκαν, εξαιτίας του γεγονότος ότι οι τιμές και των δυο μετρικών εξαρτώνται από την κατάταξη κάθε σελίδας σύμφωνα με τη δημοτικότητά της.

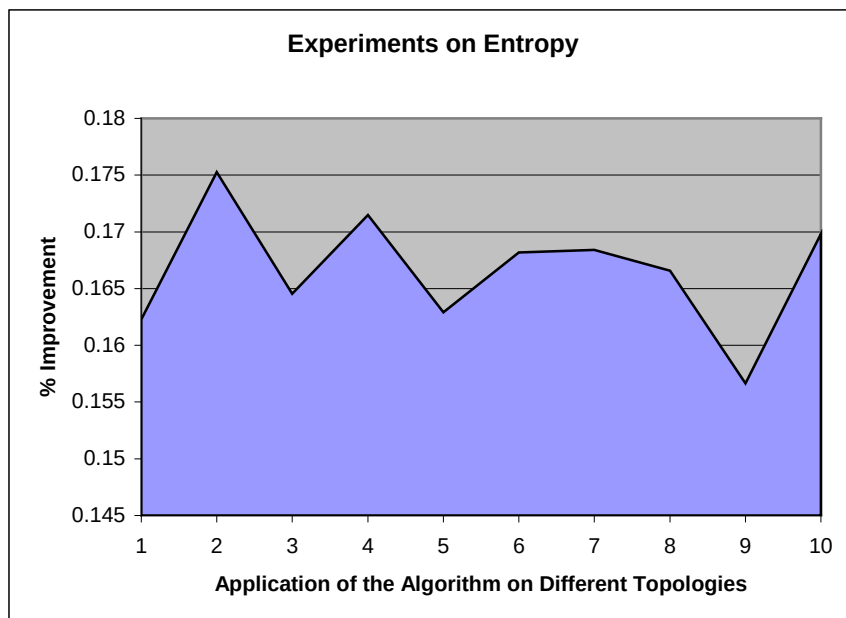


Εικόνα 24 Μέση μείωση της απόστασης από την αρχική σελίδα των, μετρημένη σε συνδέσμους

Στην γραφική παράσταση 24 παρουσιάζουμε τη μέση μείωση της απόστασης των ιστοσελίδων σε έναν ιστότοπο από την αρχική σελίδα, μετρημένη σε συνδέσμους. Στην περίπτωση τοπολογιών που αποτελούνται από 500 κόμβους η ελάχιστη μέση μείωση είναι σχεδόν 10 σύνδεσμοι, ενώ η μέγιστη μείωση είναι 160 σύνδεσμοι. Αυτό σημαίνει ότι ένας χρήστης θα μπορεί στο μέλλον να έχει πρόσβαση σε κάθε σελίδα του ιστότοπου ακολουθώντας κατά μέσο όρο 160 λιγότερους συνδέσμους. Η μεταμόρφωση του ιστότοπου σε αυτή τη περίπτωση θα είναι καταλυτική στην βελτίωση της ποιότητας της προσβασιμότητας σε αυτό, αφού πριν την εφαρμογή του αλγορίθμου ο χρήστης έπρεπε να επισκευθεί 160 περισσότερες σελίδες, που αποτελούν το 32% του συνόλου των ιστοσελίδων στην τοπολογία, για να προσπελάσει μια συγκεκριμένη ιστοσελίδα.



Εικόνα 25 Ποσοστό αύξησης της εντροπίας ενός ιστότοπου μετά την εφαρμογή του αλγορίθμου (πυκνοί γράφοι).



Εικόνα 26 Ποσοστό αύξησης της εντροπίας ενός ιστότοπου μετά την εφαρμογή του αλγορίθμου (λιγότερο πυκνοί γράφοι).

V	d	OPT	Greedy [5]	Pmin [11]	h/ph [7]	Heavypath [6]	Centipente [11]	Προτεινόμενη λύση
542	6	26.53	26.03	24.86	7.15	6.44	22.35	0.112516

Πίνακας 5 Σύγκριση προτεινόμενου αλγορίθμου με άλλες τεχνικές

Στην εικόνα 25, παρουσιάζεται πως επηρεάζει η εφαρμογή του αλγορίθμου στο γράφο ενός ιστότοπου την εντροπία του. Γενικά, η εντροπία αυξάνεται κατά ένα ποσοστό, το οποίο σε κάποιες περιπτώσεις ξεπερνά το 10% της εντροπίας πριν την εφαρμογή του αλγορίθμου. Αυτά τα αποτελέσματα είναι αναμενόμενα αλλά και ενθαρρυντικά, αφού από τη μία μεριά η εισαγωγή συνδέσμων μεταξύ των κόμβων του γράφου αυξάνει την εντροπία και από την άλλη πλευρά αυτό σημαίνει πως η συνεκτικότητα του γράφου επίσης βελτιώνεται. Επομένως, τα μονοπάτια από την αρχική σελίδα μειώνονται και η περιήγηση του χρήστη στον ιστότοπο υποβοηθείται. Η εικόνα 26 επικεντρώνεται ξανά στην εντροπία του ιστότοπου, αλλά ο αλγόριθμος εφαρμόζεται σε γράφους που είναι 40% λιγότερο πυκνοί από τους γράφους που χρησιμοποιήθηκαν για τους σκοπούς των πειραμάτων της εικόνας 25. Σε αυτήν την περίπτωση η βελτίωση της εντροπίας φτάνει το ποσοστό του 17.5%. Η μεγαλύτερη αύξηση της εντροπίας στην περίπτωση των λιγότερων πυκνών γράφων είναι φυσιολογική. Ο λόγος είναι πως ο γράφος έχει μικρότερη συνεκτικότητα και επομένως υπάρχουν περισσότεροι κόμβοι που δεν είναι καθόλου συνδεδεμένοι ή συνδέονται μέσω μεγάλων μονοπατιών, σε αντίθεση με τους πυκνούς γράφους όπου η συνεκτικότητα σημαίνει την ύπαρξη περισσότερων μονοπατιών. Αυτός είναι ο λόγος που στην πρώτη περίπτωση η ανάθεση hotlinks είναι περισσότερο ουσιαστική και παρουσιάζει καλύτερα αποτελέσματα στην πράξη.

Όσον αφορά το θέμα της σύγκρισης του αλγορίθμου με άλλους αλγορίθμους ανάθεσης hotlinks, η εργασία [6] συνεισφέρει αξιολογώντας όλες τις πρόσφατες τεχνικές πειραματικά. Τα πειράματα βασίζονται σε δεντρικές δομές που εξάγονται από τους πραγματικούς ιστότοπους αλλά και από συνθετικά στιγμιότυπα. Τα τελευταία παράγονται από μία καινούρια μέθοδο που επεξεργάζεται το μέγεθος ενός ιστότοπου μέσα στο χρόνο. Με βάση το μήκος μονοπατιού, υπολογίζεται ένας αριθμός επιπλέον τιμών. Το κέρδος $g(A)$ μιας ανάθεσης A ορίζεται ως $p(0) - p(A)$, όπου p είναι το μήκος μονοπατιού. Για λόγους συγκρισιμότητας μεταξύ διαφορετικών στιγμιότυπων υπολογίζεται και το σχετικό κέρδος ως $g(A)/p(0)$. Το επίκεντρο της μελέτης είναι τα προσεγγιστικά ποσοστά που παρατηρούνται στην πράξη. Επομένως, για όλα τα στιγμιότυπα όπου δύναται να υπολογισθεί μία βέλτιστη λύση OPT, υπολογίζονται τα ποσοστά $p(A)/p(OPT)$ αλλά και $g(OPT)/g(A)$. Υπολογίστηκε το μέσο σχετικό κέρδος της εφαρμογής του προτεινόμενου αλγορίθμου και συγκρίθηκε με τα αποτελέσματα της [6] για την περίπτωση της συλλογής 542 κόμβων. Όπως παρατηρείται, η απόδοσή του είναι καλύτερη από δύο τεχνικές αλλά είναι αρκετά χειρότερη από την απόδοση που επιτυγχάνουν οι πιο πρόσφατοι αλγόριθμοι. Το γεγονός αυτό δεν είναι αποθαρρυντικό αλλά φυσιολογικό, εφόσον η φύση του αλγορίθμου και οι στόχοι του είναι διαφορετικοί από τους αντίστοιχους των άλλων αλγορίθμων, οι οποίοι δεν λαμβάνουν υπόψιν την ομοιότητα των σελίδων και επομένως η ανάθεση hotlinks είναι ευκολότερη. Με άλλα λόγια, στόχος του αλγορίθμου είναι η σύνδεση δημοφιλών αλλά και ταυτόχρονα όμοιων ιστοσελίδων, λαμβάνοντας υπόψιν τις συνθήκες και τις ανάγκες στην περίπτωση ενός πραγματικού ιστότοπου.

3.8 Συνεισφορά της Εργασίας & Συμπεράσματα

Το κεφάλαιο παρουσίασε μια πρωτότυπη προσέγγιση στην ανάθεση hotlinks. Παρουσιάστηκε ένας αλγόριθμος που λαμβάνει υπόψιν τη δημοτικότητα των σελίδων και την μεταξύ τους ομοιότητα για να προτείνει την ανάθεση η hotlinks στην περίπτωση ενός ιστότοπου με η ιστοσελίδες, με τον περιορισμό πως κάθε κόμβος προσφέρει μόνο ένα hotlink. Παρατηρείται ότι ένας ιστότοπος που χρησιμοποιεί τον αλγόριθμο επωφελείται σε θέματα ταχύτητας πλοήγησης, όπως αναμενόταν. Κυριότερος σκοπός του αλγορίθμου είναι η προώθηση των δημοφιλών σελίδων. Πράγματι, τα πειράματα αποδεικνύουν πως οι δημοφιλείς σελίδες προωθούνται όσον αφορά τη Simetric και την απόστασή τους από την αρχική σελίδα.

Στα μελλοντικά ζητήματα περιλαμβάνεται η μελέτη πρακτικών και θεωρητικών θεμάτων. Θα ήταν ενδιαφέρον να εξακριβωθεί η αποδοτικότητα του μοντέλου σε άλλους ιστότοπους από διαφορετικούς τομείς για να επιβεβαιωθεί η επιρροή της σημασιολογικής ευαισθησίας στην ανάθεση hotlink. Ο στόχος θα είναι να παρατηρηθεί η επισκεψιμότητα στις σελίδες μετά την εκτέλεση του αλγορίθμου, για να επιβεβαιωθεί ότι η τροποποίηση της τοπολογίας και η προώθηση των δημοφιλών ιστοσελίδων επιτυγχάνονται και στην πράξη. Συγκεκριμένα, έχει μεγάλη αξία να διαπιστωθεί εάν οι χρήστες χρησιμοποιούν τα καινούρια μονοπάτια που δημιουργήθηκαν ή συνεχίζουν να προτιμούν αυτά τα οποία αποτελούνται από μη όμοιες σελίδες. Επιπλέον ο μέσος όρος των συνδέσμων που ακολουθούνται για να προσπελαστεί μια σελίδα είναι μια πολύ σημαντική μετρική που δείχνει κατά πόσο η τοπολογία του ιστότοπου έχει βελτιωθεί.

Επιπλέον, από θεωρητικής πλευράς, εφόσον γίνεται αναφορά σε τυχαιοποιημένους αλγορίθμους, έχει αξία να μελετηθούν τεχνικές για να βελτιωθεί η προσβασιμότητα σε δημοφιλείς κόμβους σε συνάρτηση με την απόσταση και την συνδεσιμότητα μεταξύ όμοιων κόμβων, αλλά χωρίς να υπερφορτώνεται η χρονική πολυπλοκότητα του αλγορίθμου.

Βιβλιογραφία Κεφαλαίου

1. M. Perkowitz, O. Etzioni, Towards adaptive web sites: conceptual framework and case study, *Computer Networks* 31 (1116) (1999) 1245–1258.
2. P. Bose, E. Kranakis, D. Krizanc, M.V. Martin, J. Czyzowicz, A. Pelc, L. Gasieniec, Strategies for hotlink assignments, in: *Proceedings of the 11th Annual International Symposium on Algorithms and Computation, LNCS*, vol. 1969, 2000, pp. 23–34.
3. J. Czyzowicz, E. Kranakis, D. Krizanc, A. Pelc, M. Vargas Martin, Evaluation of hotlink assignment heuristics for improving web access, in: *Proceedings of the Second International Conference on Internet Computing (IC'2001)*, June 25th–28th, 2001, Las Vegas, NV, USA, pp. 793–799.
4. K. Douieb, S. Langerman, Near-entropy hotlink assignments, in: *Proceedings of European Symposium on Algorithms (ESA06)*, LNCS, vol. 4618, 2006, pp. 292–303.
5. O.O. Gerstel, S. Kutten, R. Matichin, D. Peleg, Hotlink enhancement algorithms for web directories: (extended abstract), in: T. Ibaraki, N. Katoh, H. Ono (Eds.), *ISAAC, Lecture Notes in Computer Science*, vol. 2906, Springer, 2003, pp. 68–77.
6. T. Jacobs, An experimental study of recent hotlink assignment algorithms, in: *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX)*, 2008, pp. 142–151.

7. T. Jacobs, Constant factor approximations for the hotlink assignment problem, in: Proceedings of the 10th Workshop on Algorithms and Data Structures (WADS 2007), Halifax, Canada, 2007, pp. 188–200.
8. E. Kranakis, D. Krizanc, S. Shende, Approximate hotlink assignment, in: Proceedings of the 12th Annual International Symposium on Algorithms and Computation, LNCS, vol. 2223, 2001, pp. 756–767.
9. R. Matichin, D. Peleg, Approximation algorithm for hotlink assignment in the greedy model, Theoretical Computer Science 383 (1) (2007) 102–110.
10. A.A. Pessoa, E.S. Laber, C. de Souza, Efficient algorithms for the hotlink assignment problem: the worst case search, in: R. Fleischer, G. Trippen (Eds.), ISAAC, Lecture Notes in Computer Science, vol. 3341, Springer, 2004, pp. 778–792.
11. A.A. Pessoa, E.S. Laber, C. de Souza, Efficient implementation of hotlink assignment algorithm for web sites, in: L. Arge, G.F. Italiano, R. Sedgewick (Eds.), ALENEX/ANALC, SIAM, 2004, pp. 79–87.
12. K. Douieb, S. Langerman, Dynamic hotlinks, in: Proceedings of the Workshop on Algorithms and Data Structures (WADS 2005), LNCS, vol. 3608, 2005, pp. 271–280.
13. D.D. Sleator, R.E. Tarjan, A data structure for dynamic trees, Journal of Computer and System Sciences 26 (3) (1985) 362–381.
14. R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, ACM Press, Addison-Wesley Longman, New York, Harlow (England), 1999.
15. M. Sahami, T.D. Heilman, A web-based kernel function for measuring the similarity of short text snippets, in: WWW'06: Proceedings of the 15th International Conference on World Wide Web, May 2006, pp. 377–386.
16. M. Levene, G. Loizou, Computing the entropy of user navigation in the web, International Journal of Information Technology and Decision Making 2 (3) (2003) 459–476.
17. Y. Han, Deterministic sorting in $O(n \log \log n)$ time and linear space, Journal of Algorithms 50 (1) (2004) 96–105.
18. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, second ed., MIT Press and McGraw-Hill, 2001, ISBN 0-262-03293-7. Section 24.1: The Bellman–Ford algorithm, pp. 588–592. Problem 24-1, pp. 614–615.
19. A. Czumaj, M. Kowaluk, A. Lingas, Faster algorithms for finding lowest common ancestors in directed acyclic graphs, Theoretical Computer Science 380 (1–2) (2007) 37–46.
20. LEDA is a C++ class library for efficient data types and algorithms. LEDA provides algorithmic in-depth knowledge in the field of graph- and network problems, geometric computations, combinatorial optimization and other. LEDA is implemented following the object-oriented approach. <<http://www.algorithmic-solutions.com/leda/index.htm>>.
21. SEO Tools: Similar Page Checker. <<http://www.webconfs.com/similar-page-checker.php>>.
22. Ch. Makris, Y. Panagis, Y. Plegas, E. Sakkopoulos, An integrated web system to facilitate personalized web searching algorithms, in: Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), Fortaleza, Ceara, Brazil, March 16–20, 2008, pp. 2397–2402.
23. C. Makris, Y. Panagis, E. Sakkopoulos, A. Tsakalidis, Category ranking for personalized search, Data and Knowledge Engineering 60 (1) (2007) 109–125.
24. D. Antoniou, E. Sakkopoulos, P. Adamopoulou, N. Tsirakis and A. Tsakalidis, “A Web Personalizing Technique using Adaptive Data Structures: the case of bursts in web visits”, Journal of Systems and Software, Volume 83, Issue 11, November 2010, Pages 2200–2210.

25. D. Antoniou, J. Garofalakis, C. Makris, Y. Panagis and E. Sakkopoulos, "Context-similarity based hotlinks assignment: Model, metrics and algorithm", Data & Knowledge Engineering (DKE), Elsevier, Vol. 69, No 4, April 2010, pp. 357-370.
26. Dimitris Antoniou, Mersini Paschou, Efrosini Surla, Athanasios Tsakalidis," A Semantic Web Personalizing Technique,The case of bursts in web visits", Semantic Web Journal, (Invited Paper).
27. D. Antoniou , E. Sakkopoulos,"A Web Recommendation Algorithm using Splay Trees: The case of burst of visits", Data & Knowledge Engineering, (έχει υποβληθεί).
28. Dimitris Antoniou, Mersini Paschou, Efrosini Surla, Athanasios Tsakalidis," A Semantic Web Personalizing Technique,The case of bursts in web visits", International Workshop on Semantic Web And Reasoning for Cultural Heritage and Digital Libraries (SWARCH-DL), in ICSC 2010.
29. M. Levene, G. Loizou, Computing the entropy of user navigation in the web, International Journal of Information Technology and Decision Making 2 (3) (2003) 459–476.



ΜΕΡΟΣ Δ: ΕΞΑΤΟΜΙΚΕΥΣΗ ΔΙΚΤΥΑΚΩΝ ΤΟΠΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΑ ΣΥΣΤΑΣΕΩΝ

Κεφάλαιο 4

Μία Τεχνική Εξατομίκευσης του Διαδικτύου Χρησιμοποιώντας Προχωρημένες Δομές Δεδομένων: Η Περίπτωση των Εξάρσεων Επισκέψεων σε Ιστοσελίδες

Σύνοψη

Η εκρηκτική αύξηση του μεγέθους και της χρήσης του Παγκόσμιου Ιστού δημιουργεί συνεχώς νέες μεγάλες προκλήσεις και ανάγκες. Η ανάγκη για την πρόβλεψη στις προτιμήσεις των χρηστών, προκειμένου να επιταχύνει και να βελτιώσει την περιήγηση, μπορεί να επιτευχθεί με την προσωποποίηση των δικτυακών τόπων. Οι αλγόριθμοι σύστασης και εξατομίκευσης αποσκοπούν στη σύσταση ιστοσελίδων προς τους χρήστες με βάση την τρέχουσα επίσκεψή τους και τα μοτίβα πλοήγησης προηγούμενων χρηστών. Το πρόβλημα που αντιμετωπίζεται είναι η περίπτωση όπου λίγες ιστοσελίδες γίνονται πολύ δημοφιλείς για σύντομα χρονικά διαστήματα και προσπελούνται πολύ συχνά σε ένα περιορισμένο χρονικό χώρο. Στόχος είναι αυτές οι εξάρσεις των επισκέψεων να διαχειριστούν και να προταθούν αυτές οι άκρως δημοφιλείς σελίδες στο μέλλον σε χρήστες που έχουν κοινά ενδιαφέροντα. Ως εκ τούτου, σε αυτό το κεφάλαιο, προτείνεται μια νέα τεχνική εξατομίκευσης με βάση προηγμένες δομές δεδομένων.

Οι δομές δεδομένων που χρησιμοποιούνται είναι το Splay Δέντρο (1) και οι δυαδικοί σωροί (2). Περιγράφονται η αρχιτεκτονική της τεχνικής, η ανάλυση του χρόνου και η πολυπλοκότητα του χώρου και αποδεικνύεται η απόδοσή της. Επίσης, συγκρίνονται θεωρητικά και πειραματικά η προτεινόμενη τεχνική με μια άλλη προσέγγιση για την επαλήθευση της αποτελεσματικότητάς της. Η προτεινόμενη λύση επιτυγχάνει $O(P^2)$ πολυπλοκότητα χώρου και τρέχει σε $k \cdot \log P$ χρόνο, όπου k είναι ο αριθμός των σελίδων και P ο αριθμός των κατηγοριών των ιστοσελίδων.

4.1 Εισαγωγή

Ο Παγκόσμιος Ιστός παρουσιάζει την ίδια εμφάνιση σε κάθε χρήστη, ανεξάρτητα από το ιστορικό περιήγησης. Ωστόσο, κατά τη διάρκεια των τελευταίων ετών, έχουν προταθεί διάφορες τεχνικές εξατομίκευσης. Αυτές οι τεχνικές εξατομικεύουν τη διαδικτυακή εμπειρία ενός χρήστη, με την συνένωση προσωπικών πληροφοριών με τις παγκόσμιες πληροφορίες, με σκοπό την αποτελεσματική προσαρμογή της εικόνας του διαδικτύου προς το χρήστη.

Η εξατομίκευση μπορεί να οριστεί ως ο σχεδιασμός, η διαχείριση και διανομή περιεχομένου με βάση τη γνώση, παρατήρηση και πρόγνωση πληροφορίας. Οι τεχνικές εξατομίκευσης διαμορφώνονται ανάλογα το άτομο, τις προτιμήσεις του και τις συνήθειές του σε ροές από «κλικ», με προσαρμοσμένο περιεχόμενο με βάση το προφίλ χρήστη.

Στον σημερινό κόσμο της υπερφόρτωσης πληροφοριών, πολλές παρόμοιες τεχνολογίες χρησιμοποιούνται ως ένας τρόπος φιλτραρίσματος και οργάνωσης των δεδομένων.

Η σωστά εκτελεσμένη εξατομίκευση της εμπειρίας του επισκέπτη κάνει το χρόνο του σε έναν ιστότοπο ή σε μια εφαρμογή, πιο παραγωγικό και ενδιαφέρον. Η εξατομίκευση μπορεί επίσης να είναι πολύτιμη για έναν οργανισμό, μια διαδικτυακή πύλη ή ένα e-κατάστημα, γιατί οδηγεί σε επιθυμητά αποτελέσματα για την επιχείρηση όπως είναι η αυξανόμενη ανταπόκριση των επισκεπτών και η προώθηση της διατήρησης των πελατών.

Δυστυχώς, η εξατομίκευση έχει παράλληλα τη δυνατότητα να αυξήσει την πολυπλοκότητα της διεπαφής του ιστότοπου και να επιφέρει αναποτελεσματικότητα στην αρχιτεκτονική. Θα μπορούσε ακόμα και να διακινδυνεύσει την αποτελεσματικότητα του μάρκετινγκ ή ακόμη χειρότερα, να χειροτερέψει την εμπειρία του χρήστη. Λίγες είναι οι επιχειρήσεις που είναι πρόθυμες να θυσιάσουν το βασικό χαρακτήρα τους για χάρη των καινοτόμων web τεχνασμάτων.

Στο παρόν κεφάλαιο επιχειρείται η ενσωμάτωση της περίπτωσης της έξαρσης επισκέψεων στην εξατομίκευση δικτυακών τόπων. Πολλές πτυχές της καθημερινής ζωής περιγράφονται από γεγονότα [1]. Ένας απροσδόκητα μεγάλο αριθμός γεγονότων που συνέβησαν μέσα σε ένα ορισμένο χρονικό διάστημα καλείται μια έξαρση (burst), δηλώνοντας ασυνήθιστες ενέργειες ή διαδικασίες. Έξαρση μπορεί να συμβεί σε πολλές καθημερινές καταστάσεις από την οικονομία μέχρι τα φυσικά φαινόμενα, όπως το εμπόριο μετοχών και τα πεφταστέρια. Ανάλογα με τη σημασία του φαινομένου ή τη διαδικασία που παρατηρήθηκε, η αποτελεσματική ανίχνευση bursts είναι κρίσιμη. Εξ ορισμού, ένα burst εξαρτάται από την χρονική περιοχή εστίασης, δηλαδή το μέγεθος του παραθύρου. Δεδομένου ότι το μέγεθος του παραθύρου δεν είναι γνωστό εκ των προτέρων, η διαδικασία ανίχνευσης των bursts πρέπει να πραγματοποιείται με βάση πολλαπλά μεγέθη παραθύρων.

Bursts συμβαίνουν και στην κυκλοφορία ενός δικτυακού τόπου και δύναται να επηρεάσουν τη λειτουργία της σε πολλές πτυχές, όπως αυτή που ακολουθεί. Καθώς όλο και περισσότερες εμπορικές επιχειρήσεις λειτουργούν online, είναι ζωτικής σημασίας να κάνουν τους ιστοχώρους τους ελκυστικούς για τους πελάτες. Ένας τρόπος για να προσελκύσουν επισκεψιμότητα στην ιστοσελίδα είναι η online διαφήμιση στις μηχανές αναζήτησης. Στην περίπτωση αυτή, εκτός από τα αποτελέσματα της αναζήτησης, μια διαφήμιση τοποθετείται ξεχωριστά στην ιστοσελίδα της μηχανής αναζήτησης. Αν ο επισκέπτης κάνει κλικ στη διαφήμιση, ο διαφημιζόμενος οφείλει να καταβάλει αντίτιμο στη μηχανή αναζήτησης. Ένα πρόβλημα που έχει προκύψει με την pay-per-click πρακτική είναι η απάτη των κλικ (Click Fraud). Κάποιος μπορεί να χρησιμοποιήσει ένα αυτοματοποιημένο πρόγραμμα ή πρόγραμμα για την προσομοίωση πολλών κλικ από ένα πρόγραμμα περιήγησης σε μια διαφήμιση. Φυσικά, ο αριθμός των κλικ πρέπει να είναι αρκετά μεγάλος, ώστε να αποφέρει ένα σημαντικό ποσό χρημάτων. Ως εκ τούτου, ένα burst από κλικ μπορεί να είναι μια ένδειξη για Click Fraud.

Σε αυτό το κεφάλαιο, μελετάται η περίπτωση του burst επισκέψεων σε μια ιστοσελίδα και το πώς κάποιος μπορεί να αποκτήσει γνώση από το γεγονός αυτό και να ενίσχυσει την εξατομίκευση των ιστοσελίδων. Ένα πρότυπο επισκέψεων ή προσβάσεων είναι «bursty», όταν εμφανίζονται με υψηλή συχνότητα για ένα περιορισμένο χρονικό διάστημα. Ειδικότερα, σε bursty περιπτώσεις, λίγες ιστοσελίδες γίνονται πολύ δημοφιλείς για σύντομα χρονικά διαστήματα και είναι προσπελαύνονται πολύ συχνά σε ένα περιορισμένο χρονικό χώρο. Τέτοια πρότυπα έχουν επίσης παρατηρηθεί σε διάφορες

εφαρμογές του Διαδικτύου και σε διάφορες μελέτες [2]. Σε περιπτώσεις μοτίβων αναζήτησης ιστού που παρατηρούνται εξάρσεις, ο χρήστης προσπαθεί να βρει συγκεκριμένα αποτελέσματα που ανήκουν σε περιορισμένες κατηγορίες ενδιαφέροντος μέσα σε σύντομο χρονικό διάστημα. Κατά συνέπεια, ένας αποτελεσματικός μηχανισμός ανάκτησης και αποθήκευσης είναι απαραίτητος για τη φύλαξη των εξατομικευμένων κατηγοριών των χρηστών και των συχνών αποτελεσμάτων.

4.2 Περιγραφή ενός Σεναρίου Burst.

Έστω ένα σύνολο από κατηγορίες ιστοσελίδων και μια σειρά από τυχαίες επισκέψεις που εκτελούνται σε όλες τις ιστοσελίδες του κάθε χρήστη. Ένα σύνολο ιστοσελίδων ορίζεται ως προτιμώμενο από τον χρήστη όταν αυτές οι ιστοσελίδες παρουσιάζουν τους μεγαλύτερους αριθμούς επισκέψεων από αυτόν με βάση τις επισκέψεις που καταγράφονται για ένα ορισμένο χρονικό διάστημα. Πιο συγκεκριμένα, για κάθε ιστοσελίδα, μετρώνται πόσες προσβάσεις έχουν μεσολαβήσει από την τελευταία φορά που δέχθηκε επίσκεψη. Αν αυτός ο αριθμός είναι επαρκής για να αναδείξει αυτή την ιστοσελίδα ως προτιμώμενη και το χρονικό διάστημα κατά το οποίο οι προσβάσεις που εκτελούνται είναι ικανοποιητικό, τότε αυτό το μοτίβο προσβάσεων θεωρείται ως μια έξαρση επισκέψεων.

Σε αυτό το σημείο, θα πρέπει να σημειωθεί ότι η αλγοριθμική προτεινόμενη λύση χειρίζεται τις ιστοσελίδες και τις κατηγορίες των ιστοσελίδων με τον ίδιο τρόπο και λειτουργεί εξίσου αποτελεσματικά και στις δύο περιπτώσεις. Ωστόσο, λόγω της υπερφόρτωσης των ιστοσελίδων, ο διαχωρισμός σε κατηγορίες προσφέρει περισσότερα πλεονεκτήματα, δεδομένου ότι βοηθά στην υποδιαίρεση του προβλήματος. Στην επόμενη παράγραφο, θα επιχειρηθεί να εξηγηθεί γιατί.

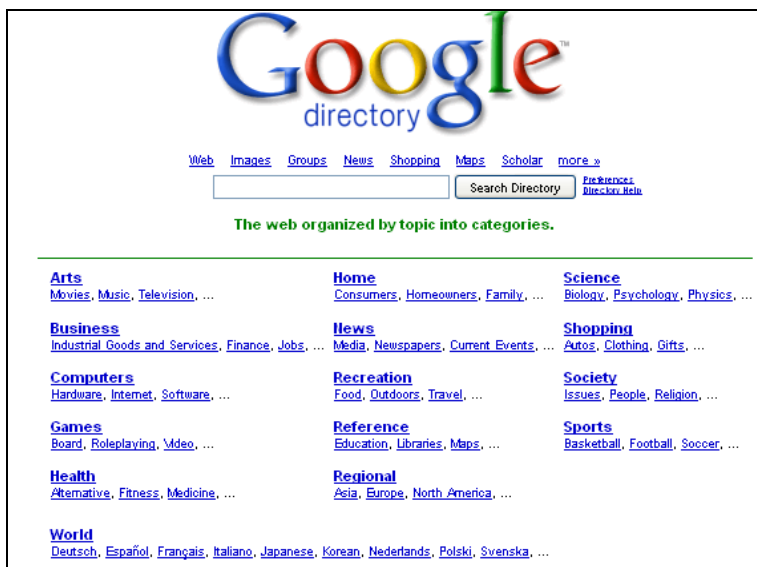
4.3 Κατηγορίες και Ιστοσελίδες

Με τη συνεχή αύξηση του αριθμού των σελίδων, έχει γίνει πολύ δύσκολο για έναν χρήστη για να εντοπίσει την επιθυμητή πληροφορία στο διαδίκτυο ή ακόμα και σε ένα δικτυακό τόπο. Για ευκολία του χρήστη, πολλοί δικτυακοί τόποι έχουν κατηγοριοποιήσει τις ιστοσελίδες τους, προκειμένου να βοηθήσουν στην αναζήτηση μιας ιστοσελίδας με την ανάθεσή της σε μια κατηγορία. Πολλά γνωστά e-shops, όπως το www.ebay.com και το www.radioshack.com έχουν οργανώσει τα προϊόντα τους σε κατηγορίες και υποκατηγορίες έτσι ώστε κάθε προϊόν να ανήκει σε ορισμένη κατηγορία και να μπορεί να προσπελαστεί ακολουθώντας το μονοπάτι που δημιουργήθηκε από την κατηγοριοποίηση των προϊόντων. Η ίδια αρχή ισχύει ακόμη και για τις μηχανές αναζήτησης. Στο επόμενο σχήμα βλέπουμε πώς η Google έχει οργανώσει το διαδίκτυο με διάφορες κατηγορίες ανά θέμα.

ΜΙΑ ΤΕΧΝΙΚΗ ΕΞΑΤΟΜΙΚΕΥΣΗΣ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΠΡΟΧΩΡΗΜΕΝΕΣ ΔΟΜΕΣ
ΔΕΔΟΜΕΝΩΝ: Η ΠΕΡΙΠΤΩΣΗ ΤΩΝ ΕΞΑΡΣΕΩΝ ΕΠΙΣΚΕΨΕΩΝ ΣΕ ΙΣΤΟΣΕΛΙΔΕΣ



Εικόνα 27 Ebay.com :Κατηγοριοποιημένη αναζήτηση



Εικόνα 28 Google.com: Το Διαδίκτυο οργανωμένο ανά θέμα σε κατηγορίες

Ως εκ τούτου, στόχος είναι να υπάρξει επωφέληση από την κατηγοριοποίηση των ιστοσελίδων και η χρησιμοποίησή της στην μελέτη των burst επισκέψεων σε μια ορισμένη κατηγορία ιστοσελίδων. Για παράδειγμα, ας υποθέσουμε ότι ένας χρήστης επισκέπτεται συνήθως την κατηγορία των ηλεκτρονικών υπολογιστών της ιστοσελίδας ενός e-shop και λόγω μιας διαδικτυακής διαφήμισης επισκέπτεται την κατηγορία του βίντεο και ήχου. Σε αυτό το σημείο, μια τεχνική εξατομίκευσης που ασχολείται με bursty επισκέψεις, θα πρότεινε στο χρήστη ιστοσελίδες και κατηγορίες ιστοσελίδων που επίσης επισκέπτονται οι χρήστες που προτιμούν την κατηγορία των βίντεο και ήχου.

4.4 Η Θεωρία της Εξατομίκευσης και Σχετικές Εργασίες

Πριν παρουσιαστούν οι συναφείς εργασίες σχετικά με το θέμα, θα γίνει μια εισαγωγή στη θεωρία εξατομίκευσης. Σε γενικές γραμμές, οι ιστοσελίδες προσαρμοσμούζονται με βάση τα ενδιαφέροντα του χρήστη. Η εξατομίκευση συνεπάγεται ότι οι αλλαγές βασίζονται σε implicit δεδομένα, όπως είδη που αγοράστηκαν ή ιστοσελίδες που αναγνώστηκαν. Υπάρχουν δύο κατηγορίες εξατομίκευσης:

1. Βασισμένη σε κανόνες
2. Με βάση το περιεχόμενο

Τα Web μοντέλα εξατομίκευσης περιλαμβάνουν το φιλτράρισμα με βάση κανόνες, με βάση την αρχή συνθηκών «αν αυτό, τότε», καθώς και το συνεργατικό φιλτράρισμα, το οποίο προσφέρει σχετικό υλικό σε πελάτες, συνδυάζοντας τις προσωπικές τους προτιμήσεις με τις προτιμήσεις άλλων ομοϊδεατών. Το συνεργατικό φιλτράρισμα αποδίδει καλά για βιβλία, μουσική, βίντεο, κ.λπ.

Υπάρχουν τρεις γενικές μέθοδοι εξατομίκευσης:

1. Άδηλη/υπονοούμενη (Implicit)
2. Ρητή (Explicit)
3. Υβρίδιο

Οι Implicit και Explicit μέθοδοι ασχολούνται με τη συγκέντρωση πληροφοριών σχετικών με το χρήστη. Η Implicit εξατομίκευση ασχολείται με τη συλλογή πληροφοριών σχετικά με τους χρήστες χωρίς να ζητά από το χρήστη πληροφορίες. Από την άλλη πλευρά, η ρητή εξατομίκευση συνήθως γίνεται ζητώντας από το χρήστη να βαθμολογήσει υπάρχοντα ή αλλαγμένα δεδομένα, να συμπληρώσει φόρμες ή να κάνει επιλογές κάνοντας κλικ σε κατάλληλους συνδέσμους κ.λπ. Η υβριδική εξατομίκευση συνδυάζει τις δύο παραπάνω προσεγγίσεις με σκοπό την ανάδειξη των πλεονεκτημάτων και των δύο.

Η προσωποποίηση του ιστού είναι ιδιαίτερα χρήσιμη σε εφαρμογές ηλεκτρονικού εμπορίου με στόχο την πρόβλεψη προτύπων συμπεριφοράς των χρηστών του Διαδικτύου και παρέχει στον αγοραστή προτάσεις για συνδέσμους, με βάση την ιστορία της περιήγησης και των αγορών. Επιπλέον, ένα αποτελεσματικό σύστημα εξατομίκευσης πρέπει να συλλέγει πληροφορίες, όπως την τοποθεσία του χρήστη, προσωπικές προτιμήσεις και στατιστικά στοιχεία ώστε να ενταχθούν στις συστάσεις που διατυπώνονται από το Web σύστημα εξατομίκευσης.

Η εξατομίκευση ιστοσελίδων έχει καταστεί ένα σημαντικό θέμα, λόγω της δημοτικότητας των εφαρμογών ηλεκτρονικού εμπορίου [3,4,5]. Διάφορες μέθοδοι για την εξατομίκευση ιστοσελίδων έχουν προταθεί [3,6,7,8]. Ο στόχος της εξατομικευμένης ιστοσελίδας είναι να επωφεληθεί από τις γνώσεις που έχουν αποκτηθεί από την ανάλυση της διαδικτυακής συμπεριφοράς του χρήστη, σε συνδυασμό με άλλες πληροφορίες που συλλέγονται, όπως την τοποθεσία του χρήστη, τα προηγούμενα μοτίβα πλοήγησης, καθώς και τα είδη που αγοράζονται [9,10,8,4,5]. Ένα άλλο πολύ σημαντικό ζήτημα είναι η δομή της ιστοσελίδας και η στατιστική μελέτη των συνδέσμων και των σελίδων από τις

οποίες αποτελείται. Ο UPR είναι ένας αλγόριθμος παρόμοιος με τον PageRank, που συνδυάζει δεδομένα χρήσης και τεχνικές ανάλυσης συνδέσμων για την ανάθεση πιθανοτήτων σε ιστοσελίδες με βάση τη σημασία τους στο γράφημα πλοήγησης της τοποθεσίας Web [12].

Μια άλλη τεχνική που χρησιμοποιείται ευρέως για εξατομίκευση είναι η εξόρυξη δεδομένων από τη χρήση του διαδικτύου[3]. Για παράδειγμα, ένας αλγόριθμος κατηγοριοποίησης για εξατομίκευση που βασίζεται σε τεχνικές εξόρυξης λαμβάνει υπόψη τόσο στατικές πληροφορίες, μέσω κλασσικών τεχνικές ομαδοποίησης, αλλά και τη δυναμική συμπεριφορά του χρήστη, με αποτέλεσμα να προτείνει ένα καινοτόμο και αποτελεσματικό αλγόριθμο επανακατηγοριοποίησης [13].

Υπολογιστικοί ευφυείς συνδυασμοί έχουν επίσης εφαρμοστεί στο πλαίσιο της εξατομίκευσης, παρέχοντας διάφορα παραδείγματα από ευφυή συστήματα που έχουν σχεδιαστεί για να παρέχουν στους χρήστες του Παγκόσμιου Ιστού τις πληροφορίες που αναζητούν, χωρίς αυτές να ζητηθούν ρητά [14].

Από την άλλη πλευρά, ένα καινοτόμο online σύστημα συστάσεων χτίζει μοντέλα των προφίλ των χρηστών και προσφέρει προτάσεις, χωρίς ο χρήστης να αναλαμβάνει την πρωτοβουλία [15]. Σε γενικές γραμμές, η εξατομίκευση πρόσφατα έχει χρησιμοποιηθεί και σε πολλούς άλλους τομείς. Στη διαφήμιση, μία νέα online τεχνική διαφημιστικής στόχευσης προτείνεται [16], η οποία προσαρμόζει και χρησιμοποιεί διάφορες ισχυρές και καλά δοκιμασμένες τεχνικές ανάκτησης πληροφορίας ή λεξιλογικές για να αναπτύξει μια εκτίμηση της συγγένειας ενός χρήστη με συγκεκριμένα προϊόντα και υπηρεσίες με βάση την ανάλυση της συμπεριφοράς ενός χρήστη στην πλοήγηση στον παγκόσμιο ιστό. Επιπλέον, προτείνεται μια ερευνητική σχεδιαστική προσέγγιση που συνδυάζει τη συμπεριφορά με τεχνικές μηχανικής για την καλύτερη μοντελοποίηση χρήστη σε εξατομικευμένες φορητές εφαρμογές διαφήμισης [17]. Ακόμη και στην λήψη φορητών παιχνιδιών μέσω κινητής συσκευής, προτείνεται ένα εξατομικευμένο φορητό σύστημα σύστασης παιχνιδιών, το οποίο λαμβάνει υπόψη το χρόνο της ημέρας, ενώ η ώρα της εβδομάδας χρησιμοποιείται για να παρέχει μια πιο εξατομικευμένη εμπειρία [18]. Όσον αφορά τα bursts ως αλγοριθμικό θέμα, έχουν επίσης παρουσιαστεί διάφορες εργασίες. Ένα νέο πλαίσιο για την αλγοριθμική ελαστική ανίχνευση bursts έχει προταθεί: μια οικογένεια δομών δεδομένων που γενικεύει το Shifted Binary Tree, και ένας ευρετικός αλγόριθμος αναζήτησης για την εύρεση μιας αποτελεσματικής δομής δεδομένης της εισόδου [19].

4.5 Εισαγωγή στις Δομές Δεδομένων για εξατομίκευση

Προτείνεται μια τεχνική εξατομίκευσης, η οποία χρησιμοποιεί προηγμένες προσαρμοζόμενες δομές δεδομένων, με σκοπό την καταγραφή της επίσκεψης κάθε χρήστη σε μια συγκεκριμένη ιστοσελίδα ή κατηγορία ιστοσελίδων και με βάση τις πληροφορίες που συγκεντρώθηκαν από τις επισκέψεις σε έναν ιστότοπο, προτείνει στον χρήστη άλλους ενδεχομένως ενδιαφέροντες ή χρήσιμους συνδέσμους. Στόχος είναι, κάθε φορά που ένας χρήστης κάνει κλικ σε μια σελίδα που ανήκει σε ορισμένη κατηγορία A, στην οποία έχει παρατηρηθεί μια έξαρση επισκέψεων από αυτόν τον χρήστη, να προτείνεται μια σειρά από τις πιο δημοφιλείς κατηγορίες, τις οποίες οι περισσότεροι χρήστες που επισκέφτηκαν την A, προτιμούν επίσης να επισκέπτονται. Οι δομές

δεδομένων που χρησιμοποιούνται είναι το splay δέντρο (1) [20] και οι δυαδικοί σωροί (Binary Heaps) (2) [21,22]. Στις επόμενες παραγράφους δίνεται μια σύντομη περιγραφή των εν λόγω δομών δεδομένων.

4.5.1 Splay δέντρο

Τα splay δέντρα εκτελούν πολλές περιστροφές μετά από κάθε πρόσβαση, πράγμα που τα καθιστά λιγότερο αποτελεσματικά στην πράξη σε ορισμένες εφαρμογές. Αυτές οι περιστροφές μπορεί να είναι ιδιαίτερα επιβλαβείς, όταν οι κόμβοι έχουν επανξηθεί με βοηθητικές δομές. Παρ'όλα αυτά, τα splay δέντρα χρησιμοποιούνται έτσι ώστε να αξιοποιηθούν οι δυνατότητές τους που προκύπτουν από τις υπό όρους παραλλαγές του splaying και να ενσωματωθούν στον υπό μελέτη αλγόριθμο. Σε γενικές γραμμές, τα splay δέντρα έχουν μοναδικές ιδιότητες που τα καθιστούν περισσότερο από κατάλληλα για την επίλυση του προβλήματος που αντιμετωπίζεται σε αυτό το κεφάλαιο, διότι εκμεταλλεύονται την τοπικότητα χώρου και χρόνου.

Η τοπικότητα δεδομένων είναι πολύ πιο εύκολο να διατηρηθεί σε προβλήματα που έχουν στατικά δεδομένα. Αλλά για δυναμικά δεδομένα, όπως συμβαίνει στα splay δέντρα, η διατήρηση της τοπικότητας απαιτεί σημαντική προσπάθεια. Σε δυναμικά προβλήματα, όπως αυτό της σύστασης ορισμένων ιστοσελίδων για τους χρήστες, ο σχεδιασμός της δομής των δεδομένων είναι ζωτικής σημασίας για το πρόβλημα της αποτελεσματικής μετακίνησης δεδομένων. Στα splay δέντρα τα πιο πρόσφατα προσπελασμένα στοιχεία μεταφέρονται πιο κοντά στη ρίζα και το splay δέντρο είναι δυναμικά αυτοοργανώσιμο σύμφωνα με τα στοιχεία πρόσβασης, έτσι ώστε τα λιγότερο συχνά χρησιμοποιούμενα στοιχεία να είναι τα πιο απομακρυσμένα από την ρίζα.

Ένα καλό παράδειγμα εφαρμογής, στην οποία παρατηρείται η περίπτωση όπου το ίδιο στοιχείο αναζητείται πολλές φορές είναι ένας δρομολογητής (router) [23]. Οι δρομολογητές πρέπει να αποφασίσουν σε ποιο εξερχόμενο σύρμα να δρομολογήσουν τα εισερχόμενα πακέτα, με βάση την διεύθυνση IP. Ο δρομολογητής χρειάζεται ένα μεγάλο πίνακα (ένα χάρτη), που να μπορεί να χρησιμοποιηθεί για να αναζητηθεί μια διεύθυνση IP και ανακαλύψει την εξερχόμενη σύνδεση που θα χρησιμοποιήσει. Αν μια διεύθυνση IP έχει χρησιμοποιηθεί μία φορά, είναι πιθανό να χρησιμοποιηθεί και πάλι και ίσως πολλές φορές. Τα splay δέντρα έχουν σχεδιαστεί για να παρέχουν καλή απόδοση σε αυτήν την κατάσταση.

Όσον αφορά τις λεπτομέρειες υλοποίησης, πέρα από την ευκολία στον προγραμματισμό και την αποτελεσματικότητα στην πράξη, τα splay δέντρα επίσης χρησιμοποιούν λιγότερο χώρο, διότι δε διατηρούνται πληροφορίες ζύγισης. Ωστόσο, τα μειονεκτήματά τους είναι ότι απαιτούν προσαρμογές σε κάθε αναζήτηση, με αποτέλεσμα να προκύπτουν δαπανηρές μεμονωμένες ενέργειες. Βέβαια, στην περίπτωση μιας εξαιρετικά δημοφιλούς ιστοσελίδας, οι μεμονωμένες πράξεις δεν είναι συνήθεις και ως εκ τούτου αυτό το πεδίο είναι ιδανικό για την εφαρμογή των splay δέντρων.

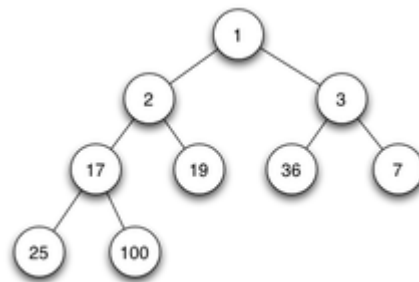
4.5.2 Δυαδικός Σωρός (Binary Heap)

Ένα δυαδικό σωρός [21,22] είναι μία δομή δεδομένων σωρού που δημιουργήθηκε χρησιμοποιώντας ένα δυαδικό δέντρο. Μπορεί να θεωρηθεί ως ένα δυαδικό δένδρο με δύο επιπλέον περιορισμούς:

Ιδιότητα σχήματος: το δένδρο είναι ένα πλήρες δυαδικό δέντρο, δηλαδή όλα τα επίπεδα του δέντρου, εκτός ίσως από το τελευταίο, είναι πλήρως συμπληρωμένα και εάν το τελευταίο επίπεδο του δέντρου δεν είναι πλήρες, οι κόμβοι αυτού του επιπέδου, είναι συμπληρωμένοι από τα αριστερά προς τα δεξιά.

Η ιδιότητα σωρού: κάθε κόμβος είναι μεγαλύτερος ή ίσος από κάθε ένα από τα παιδιά του, σύμφωνα με κάποιο ορισμένο κατηγορήμα σύγκρισης, το οποίο είναι σταθερό για όλη τη δομή δεδομένων.

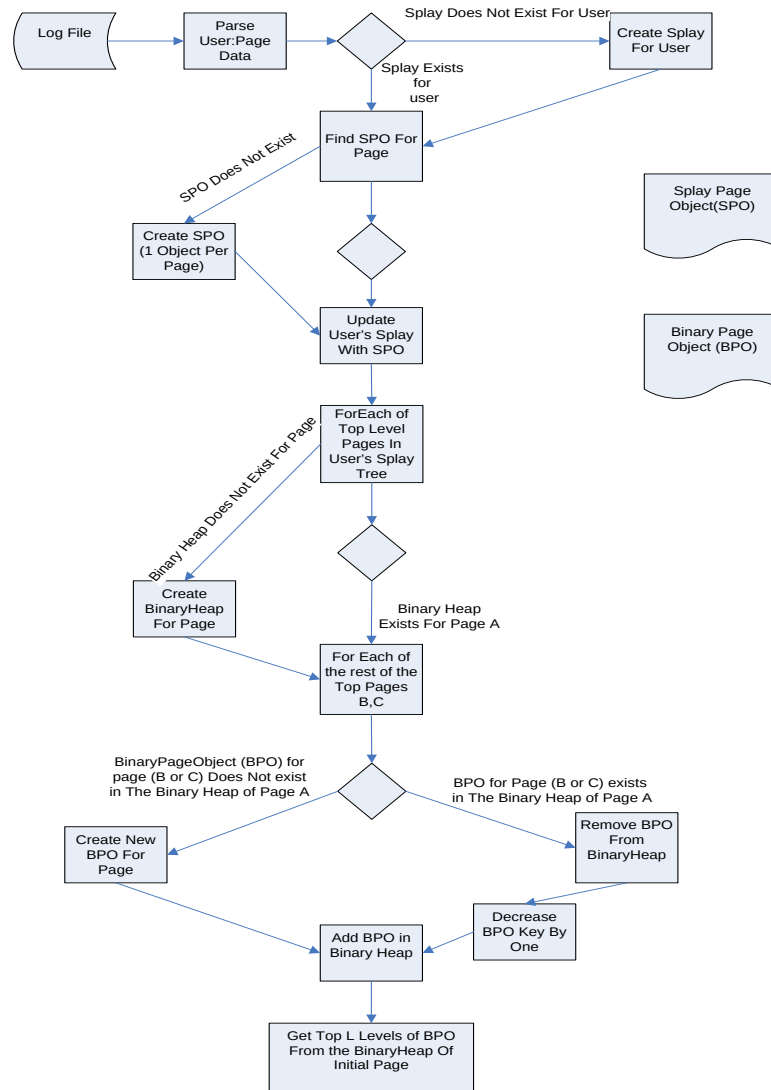
Ο δυαδικός σωρός είναι μια ειδική περίπτωση του d-αδικού σωρού, στον οποίο κάθε κόμβος μπορεί να έχει d παιδιά (και όχι μόνο 2). Η ιδιότητα του δυαδικού σωρού που αξιοποιείται, είναι το γεγονός ότι η επιστροφή ενός δείκτη προς τον κόμβο που περιέχει το ελάχιστο στοιχείο στο σωρό χωρίς να το αφαιρεί, έχει κόστος $O(1)$ χρόνο.



Εικόνα 29 Παράδειγμα Δυαδικής Σωρού

4.6 Η προτεινόμενη λύση: Εξατομίκευση στην περίπτωση σεναρίου εξάρσεων επισκέψεων

Το πρόβλημα περιγράφεται ως εξής: Έστω ένα σύνολο P κατηγοριών ιστοσελίδων και N χρήστες. Κάθε ιστοσελίδα ανήκει σε κάποια κατηγορία και το προφίλ του κάθε χρήστη πρέπει να φυλάσσεται σε ένα splay δέντρο. Ως προφίλ, ορίζουμε το "logfile" των ιστοσελίδων που ο χρήστης έχει επισκεφθεί μέχρι τώρα. Στο splay δέντρο αποθηκεύονται οι κατηγορίες των ιστοσελίδων. Σύμφωνα με τις ιδιότητες του splay δέντρου, το στοιχείο που δέχθηκε την τελευταία επίσκεψη έρχεται στη ρίζα του δέντρου. Στην προκειμένη περίπτωση, το δέντρο μπορεί να τροποποιηθεί έτσι ώστε το στοιχείο που δέχεται τις συχνότερες επισκέψεις να είναι εκείνο που μεταφέρεται στη ρίζα. Στην πραγματικότητα, στόχος είναι να εφαρμοστεί splay σε μια κατηγορία όταν παρατηρηθεί σε αυτή μια έξαρση επισκέψεων. Αφότου μια κατηγορία έρθει στη ρίζα λόγω μιας έξαρσης επισκέψεων, δεν απαραίτητη άλλη ανακατασκευή ή splaying. Ως εκ τούτου, οι κατηγορίες που εμφανίζονται στα ανώτερα επίπεδα του δέντρου του κάθε χρήστη είναι αυτές που προτιμά ο χρήστης.



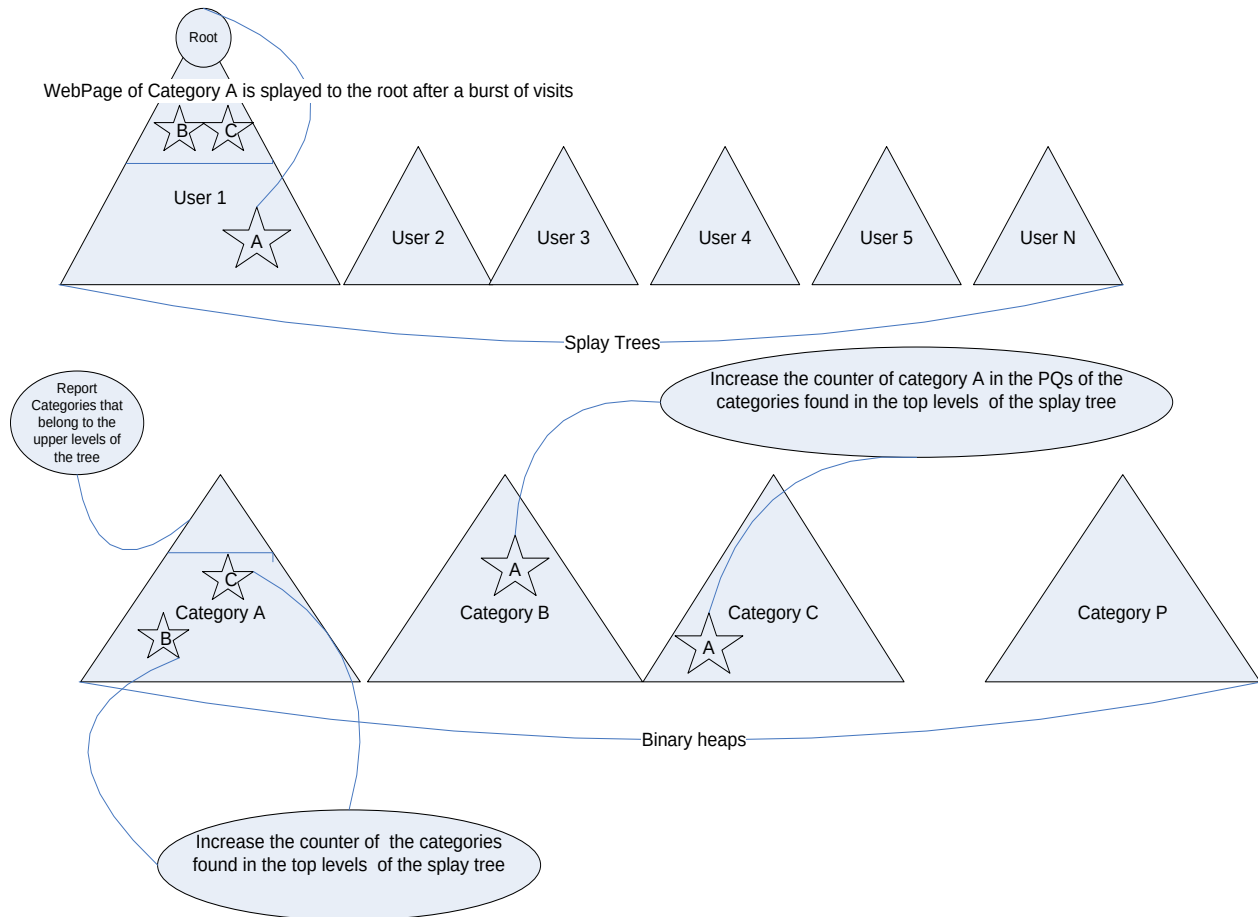
Εικόνα 30 Διάγραμμα ροής προτεινόμενου αλγορίθμου

Δεδομένου ότι διατηρείται ένα προφίλ για κάθε χρήστη, σκοπός είναι να οικοδομηθεί μια δομή δεδομένων που θα κρατήσει τις πιο δημοφιλείς κατηγορίες τις οποίες οι περισσότεροι χρήστες που επισκέφτηκαν την A, για παράδειγμα, προτιμούν επίσης να επισκέπτονται. Επομένως, για κάθε κατηγορία κτίζεται μια ουρά προτεραιότητας. Η ουρά προτεραιότητας που έχει επιλεγεί για τους σκοπούς της παρούσας διατριβής είναι ο δυαδικός σωρός. Σε κάθε κατηγορία αντιστοιχεί ένας δυαδικός σωρός με τις υπόλοιπες κατηγορίες και τις δημοτικότητές τους. Σύμφωνα με τις ιδιότητες της ουράς προτεραιότητας, στη ρίζα βρίσκεται το ελάχιστο κλειδί και ως εκ τούτου μπορεί να προσπελαστεί σε $O(1)$ χρόνο. Αποθηκεύεται ο αριθμός της δημοτικότητας της κάθε κατηγορίας με αρνητικό πρόσημο, με σκοπό να διατηρείται η

μέγιστη στη ρίζα της ουράς προτεραιότητας. Κάθε φορά που παρατηρείται μια έξαρση επισκέψεων σε μια κατηγορία A , θα αυξηθεί κατά ένα ο μετρητής της δημοτικότητας αυτής της κατηγορίας σε όλες τις ουρές προτεραιότητας των κατηγοριών που βρέθηκαν στα κορυφαία επίπεδα του splay δέντρου του συγκεκριμένου χρήστη. Επιπλέον, αυξάνονται οι μετρητές αυτών των κατηγοριών στην ουρά προτεραιότητας της κατηγορίας A . Έτσι, είναι δυνατό να εξαχθεί η πιο δημοφιλής κατηγορία για τα άτομα που επισκέφτηκαν την κατηγορία A σε σταθερό χρόνο.

Ανάλογα με το αν στα splay δέντρα αποθηκεύονται κατηγορίες ιστοσελίδων ή ιστοσελίδες, μπορεί να υπάρξει μια διαφορετική προσέγγιση. Στην περίπτωση των ιστοσελίδων, με σκοπό να εξασφαλιστεί ότι τουλάχιστον ένα μεγάλο μέρος των σελίδων της κατηγορίας είναι ανεβαίνουν μέσω splay στη ρίζα, μπορούμε να ακολουθηθεί η εξής τεχνική. Έστω μια σελίδα x έχει προσπελαστεί k φορές, οι οποίες είναι αρκετές για να χαρακτηριστεί αυτή τη σελίδα ως η πιο συχνά προσπελάσιμη σελίδα. Στη συνέχεια, εφόσον ο πατέρας του παρόντος κόμβου ανήκει στην ίδια κατηγορία, διατρέχεται το δέντρο από κάτω προς τα πάνω. Το άνω όριο του αριθμού των επιπέδων που διατρέχεται προς τα πάνω, εξαρτάται από τον αριθμό των κατηγοριών και ιστοσελίδων. Έστω z ο τελευταίος πρόγονος του x που ανήκει στην ίδια κατηγορία με το x . Τότε, εφαρμόζεται splay σε όλους τους κόμβους που ανήκουν στο υποδέντρο. Επιπλέον, πρέπει να διακρίνονται οι κατηγορίες που δέχονται τις περισσότερες επισκέψεις από κάθε χρήστη. Οι κατηγορίες πιο κοντά στη ρίζα του δέντρου, είναι εκείνες που προτιμά ο χρήστης. Ως εκ τούτου, το μόνο πράγμα που απομένει να αποφασιστεί είναι το βάθος του δέντρου που αντιπροσωπεύει το όριο, πέραν του οποίου όλες οι κατηγορίες θεωρούνται προτιμώμενες από το χρήστη.

Από την πλευρά του χρήστη, ο αλγόριθμος θα λειτουργεί ως εξής. Έστω πως η ιστοσελίδα V ενός συγκεκριμένου Τμήματος που υπάγεται στο πανεπιστήμιο δέχεται πολλές επισκέψεις σε μια σύντρομη χρονική περίοδο T , λόγω της αναγγελίας των βαθμών ενός μαθήματος. Τα μεγέθη των V και T είναι ικανοποιητικά και επομένως παρουσιάζεται ένα burst και ο αλγόριθμος εκτελείται. Ως αποτέλεσμα, η bursty ιστοσελίδα θα προταθεί από τον αλγόριθμο στον επόμενο φοιτητή που επισκέπτεται την ιστοσελίδα του Τμήματος,



Εικόνα 31 Παράδειγμα εκτέλεσης του αλγορίθμου στο επίπεδο των δομών δεδομένων

```

1. While (A webpage of category W is accessed by a user)
   /*Retrieve information from the Webpage's logfile*/
2. if (this access has created a burst of visits to the category W)
3. then
   /* Rearrange the splay tree of the user, so that the category with the
   latest bursty access pattern is brought to the root */
4. Splay the category W to the root of the user's splay tree
   /* Update the priority queues of the categories in order to promote the
   popular categories of the user */
5. Define the set of categories ,TOP, that already exist in the top levels
of the splay tree
6. Increase the counter of W in all the priority queues of the categories
that belong to TOP
    
```

```
7. Increase the counter of the categories that belong to TOP in the
priority queue of W
8. Return as suggested the root of the priority queue of W
9. endif
10.      else
11.      continue
```

4.7 Η Ανάλυση χώρου και χρόνου

4.7.1 Η λύση με Πίνακες

Για να διαπιστωθεί η αποτελεσματικότητα του αλγορίθμου, περιγράφεται πρώτα μια απλοϊκή προσέγγιση του προβλήματος, η οποία βασίζεται στη χρήση των πινάκων. Ο λόγος για τον οποίο επιλέγεται αυτό το είδος προσέγγισης είναι ότι αυτή θα ήταν η προφανής και απλούστερη λύση, σε περίπτωση που το πεδίο εκτέλεσης είναι ένα απλό μοντέλο RAM ή πρέπει να χρησιμοποιηθούν τα δεδομένα κίνησης στον ιστότοπο που είναι αποθηκευμένα σε μια βάση δεδομένων. Η προτεινόμενη λύση είναι αποδεδειγμένα καλύτερη από άποψη χρόνου καθώς και χώρου. Έστω η περίπτωση που αποτελείται από W σελίδες, P κατηγορίες και N χρήστες. Για το προφίλ του κάθε χρήστη χρειάζεται ένα διάνυσμα μεγέθους W . Επομένως, συνολικά, χρειάζονται N τέτοια διανύσματα και ο χώρος που απαιτείται είναι $N \cdot W + N \cdot P$, δεδομένου ότι χρειάζεται επίσης ένα διάνυσμα για κάθε χρήστη, όπου αποθηκεύεται η κατηγορία, στην οποία ανήκει κάθε σελίδα επίσκεψης.

Έστω k ο αριθμός των σελίδων που αντιπροσωπεύουν τα αγαπημένα του κάθε χρήστη. Έτσι, όσον αφορά το χρόνο, ανά πρόσβαση, απαιτείται:

- $W \cdot \log W$, για να ταξινομηθούν οι σελίδες σύμφωνα με τη δημοτικότητά τους, χρησιμοποιώντας για παράδειγμα τον αλγόριθμο Mergesort που έχει $O(n \log n)$ χρονική πολυπλοκότητα για n στοιχεία.
- $P \cdot \log P$, για να ταξινομηθούν οι κατηγορίες ανάλογα με την δημοτικότητά τους.
- Και τέλος, $k \cdot N$, προκειμένου να επιστραφούν οι k πρώτες θέσεις του διανύσματος του καθενός από τους N χρήστες.

4.7.2 Η Προτεινόμενη Λύση

4.7.2.1 Απαιτήσεις σε Χώρο

Όσον αφορά την πολυπλοκότητα χώρου, ο χώρος, όπως αναμενόταν, κυρίως καταναλώνεται από τις δύο δομές δεδομένων.

- Splay δέντρα: Χρειαζόμαστε ένα splay δέντρο για κάθε χρήστη. Στην χειρότερη περίπτωση, σε κάθε splay δέντρο αποθηκεύονται W ιστοσελίδες. Επομένως, λαμβάνοντας υπ' όψιν τα πρόσθετα πεδία που απαιτούνται για κάθε κόμβο του δένδρου, ο απαιτούμενος χώρος είναι $5 \cdot N \cdot W$.
- Ουρές προτεραιότητας: Για κάθε κατηγορία απαιτείται μια ουρά προτεραιότητας. Ως εκ τούτου, για P κατηγορίες, χρειάζεται $O(P^2)$ χώρο.

4.7.2.2 Απαιτήσεις σε Χρόνο

Όσον αφορά την πολυπλοκότητα χρόνου, ανά πρόσβαση χρειάζεται:

- $\tau \cdot \log\left(\frac{\tilde{W}}{W}\right)$, προκειμένου να εφαρμοστεί splay σε τ κόμβους. Αυτό είναι το πολύ $\tau \cdot \log(\# \text{ σελίδες})$.
- Χρειάζεται $O(1)$ χρόνος για να επιστραφεί η ρίζα από κάθε ουρά προτεραιότητας. Ως εκ τούτου, απαιτείται $N \cdot O(1)$ χρόνος για να προταθεί μία κατηγορία σε N χρήστες.

Τέλος, πρέπει να ενημερωθούν οι ουρές προτεραιότητας. Με άλλα λόγια, προτού προταθεί η ρίζα της ουράς προτεραιότητας μιας κατηγορίας, πρέπει να αυξηθούν τα κλειδιά των κατηγοριών που ανήκουν στις αγαπημένες του χρήστη, εάν υπάρχουν ήδη στην ουρά προτεραιότητας της κατηγορίας στην οποία εφαρμόζεται splay. Τέλος, πρέπει να αυξηθεί το κλειδί της κατηγορίας, στην οποία εφαρμόζεται splay, σε όλες τις ουρές προτεραιότητας των κορυφαίων επιπέδων του δένδρου του χρήστη. Έτσι, συνολικά χρειάζεται $k \cdot \text{Log}P$ χρόνος.

4.8 Πειράματα

Σε ό, τι αφορά τα πειράματα, προκειμένου να δοκιμαστεί ο αλγόριθμος υλοποιήθηκε η λύση των πινάκων και η προτεινόμενη λύση και συγκρίθηκαν από την άποψη του χρόνου και της αποτελεσματικότητας. Το επιλεγμένο περιβάλλον υλοποίησης είναι η αντικειμενοστραφής γλώσσα προγραμματισμού Java. Τα πειράματα έγιναν σε ένα Pentium (R), IV 3,00 GHz σύστημα με μνήμη RAM 1 GB και με λειτουργικό σύστημα. Microsoft Windows XP SP3. Η πλατφόρμα που χρησιμοποιήθηκε για την υλοποίηση είναι η NetBeans IDE 6.8. Επιπλέον έπρεπε να επιλεγθεί ένα σύνολο δεδομένων που θα χρησιμοποιηθεί ως είσοδος των αλγορίθμων. Λαμβάνοντας υπόψιν το γεγονός ότι δεν χρειάζεται μόνο ένας μεγάλος αριθμός ιστοσελίδων, αλλά αυτές οι σελίδες πρέπει να

ανήκουν και σε συγκεκριμένη κατηγορία, το σύνολο δεδομένων που επιλέχθηκε είναι το "4 universities dataset».

4.8.1 Το Σύνολο Δεδομένων «4 πανεπιστήμια»

Αυτό το σύνολο δεδομένων περιλαμβάνει WWW-σελίδες που συλλέχθηκαν από τις υπηρεσίες επιστήμης των υπολογιστών των διαφόρων πανεπιστημίων τον Ιανουάριο του 1997 από το έργο «World Wide Knowledge Base (Web->Kb)» της ομάδας μάθησης κειμένου CMU. Οι 8.282 σελίδες κατατάσσονται στις ακόλουθες κατηγορίες:

Σπουδαστές	(1641)
Σχολή	(1124)
Προσωπικό	(137)
Τμήμα	(182)
Μάθημα	(930)
Έργο	(504)
Άλλα	(3764)

Για κάθε κατηγορία το σύνολο δεδομένων περιέχει σελίδες από τα τέσσερα πανεπιστήμια
Cornell (867)
Τέξας (827)
Ουάσιγκτον (1205)
Wisconsin (1263)
και 4.120 διάφορες σελίδες που συλλέχθηκαν από άλλα πανεπιστήμια.

Τα δεδομένα είναι διαθέσιμα στο σύνδεσμο
<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/webkb-data.gtarg.gz>
Ως εκ τούτου, έχουν προκύψει $7 * 5 = 35$ κατηγορίες ιστοσελίδων. Στη συνέχεια, μετά από κανονικοποίηση, προέκυψαν 1360 ιστοσελίδες, οι οποίες είναι αυτές που ανήκουν σε κατηγορίες με 200 ιστοσελίδες κατ'ανώτατο όριο η καθεμία.

4.8.2 Περιγραφή των πειραμάτων

Πριν από την παρουσίαση και την αξιολόγηση των αποτελεσμάτων των πειραμάτων, θα πρέπει να δοθεί μια σύντομη περιγραφή της μεθοδολογίας που ακολουθείται. Πρώτα απ' όλα, όλα τα πειράματα διεξήχθησαν με βάση το ίδιο μοτίβο επισκέψεων. Το πρότυπο των επισκέψεων εκτελέστηκε για τη λύση των πινάκων και καταγράφηκαν οι προτεινόμενες ιστοσελίδες και ο χρόνος που απαιτείται μετά από κάθε επίσκεψη, προκειμένου να προταθεί μια ιστοσελίδα για το χρήστη. Το επόμενο βήμα είναι η εκτέλεση της προτεινόμενης λύσης πάνω στο ίδιο πρότυπο επισκέψεων. Και σε αυτή την περίπτωση, καταγράφονται τα αποτελέσματα και καθώς και το χρόνο εκτέλεσης. Στις επόμενες παραγράφους, συγκρίνεται η αποδοτικότητα και η αποτελεσματικότητα και των δύο λύσεων.

Κύριο μέλημά, εκτός από το χρόνο που απαιτείται για να προταθεί μια ιστοσελίδα στο χρήστη, είναι κατά πόσον η προτεινόμενη ιστοσελίδα είναι εκείνη που αναμένεται

και στην περίπτωση αυτή, αν ταιριάζει με αυτήν που προτείνεται από τη λύση των πινάκων. Ως εκ τούτου, πρέπει να μετρηθεί το ποσοστό των ιστοσελίδων που προτείνονται από την προτεινόμενη λύση και συμφωνούν με εκείνες της λύσης των πινάκων. Στις πειραματικές μελέτες που πραγματοποιήθηκαν στην παρούσα εργασία, χρησιμοποιήθηκαν κοινώς αποδεκτές μετρήσεις απόδοσης υποθέτοντας ότι αν ένα προβλεπόμενο σχόλιο ταιριάζει ακριβώς με ένα σχόλιο αναφοράς, είναι πραγματικό θετικό, ειδάλλως είναι μια ψευδώς θετικά. Οποιοδήποτε σχόλιο αναφοράς που δεν έχει καμία αντίστοιχη πρόβλεψη είναι ψευδώς αρνητικό.

Έτσι, στις υπό μελέτη περιπτώσεις, το ενδιαφέρον έγκειται στην αξιολόγηση της ποιότητας των απαντήσεων που επιστρέφονται από το σύστημα σύμφωνα με τις ακόλουθες μετρικές:

- Η ανάκληση είναι ένα μέτρο του πόσο καλά αποδίδει το εργαλείο στην εύρεση σχετικών πληροφοριών: ο αριθμός των εντοπισμένων σχετικών οντοτήτων διαιρείται με τον αριθμό όλων των σχετικών οντοτήτων στη συλλογή. (βλέπε τύπο (1))

- Η ακρίβεια είναι ένα μέτρο του πόσο καλά το εργαλείο αποδίδει στη μη επιστροφή άσχετων πληροφοριών: ο αριθμός των εντοπισμένων σχετικών οντοτήτων διαιρείται με τον αριθμό όλων των οντοτήτων. (βλέπε τύπο (2))

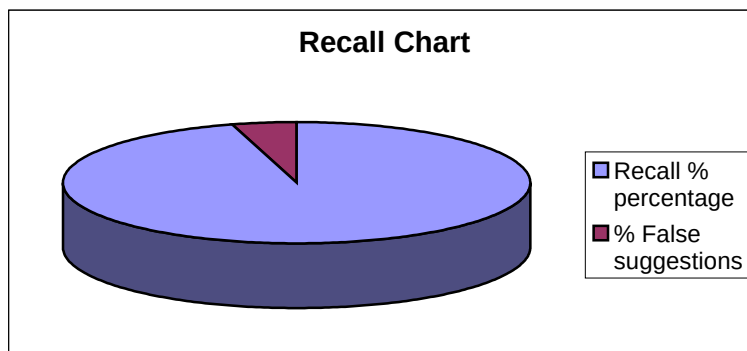
Με σκοπό να απεικονιστούν και να συγκριθούν τα αποτελέσματα της προτεινόμενης λύσης με εκείνες που βασίζονται στη χρήση των πινάκων, χρησιμοποιούνται η ακρίβεια και η ανάκληση. Έστω ένα αίτημα ερωτήματος Q_i και η προκύπτουσα δέσμη των προτεινόμενων εγγράφων R από τη λύση των πινάκων (δηλαδή των προτεινόμενων συνδέσμων με βάση τον αλγόριθμο που χρησιμοποιεί πίνακες). Επομένως, $|R|$ είναι ο αριθμός των εγγράφων/URLs σε αυτό το σύνολο. Ας υποθεθεί ότι σύμφωνα με ένα συγκεκριμένο σενάριο, η προτεινόμενη λύση προτείνει ένα σύνολο απαντήσεων A , και επομένως $|A|$ είναι ο αριθμός αυτών των εγγράφων-URLs. Η ανάκληση και η ακρίβεια ορίζονται ως:

$$\text{Ανάκληση: } \frac{|R \cap A|}{|R|} \quad (1)$$

$$\text{Ακρίβεια: } \frac{|R \cap A|}{|A|} \quad (2)$$

Όσον αφορά την αποτελεσματικότητα της προτεινόμενης λύσης, μετά την εκτέλεση του αλγορίθμου και της λύσης με τους πίνακες πάνω στα δεδομένα που έχουν ήδη περιγραφεί, έγινε σύγκριση των δύο μετρικών, της ανάκλησης και της ακρίβειας, με σκοπό να συγκριθούν τα αποτελέσματα των δύο τεχνικών. Τα αποτελέσματα δείχνουν πράγματι ότι ο προτεινόμενος αλγόριθμος λειτουργεί σωστά και προτείνει τις αναμενόμενες ιστοσελίδες. Με άλλα λόγια, η μεγάλη πλειοψηφία των ιστοσελίδων που επιστρέφονται από τον αλγόριθμο ταιριάζει με αυτές που προτείνει η λύση με τους πίνακες. Εκτελέστηκαν διάφορα μοτίβα προσβάσεων προς τις ιστοσελίδες και τις κατηγορίες του συνόλου δεδομένων και σε όλες τις περιπτώσεις οι τιμές της Ανάκλησης και της Ακρίβειας είναι κοντά στο ένα. Αυτό σημαίνει ότι πάνω από το 95% των ιστοσελίδων που προτείνονται σε κάθε πείραμα ταιριάζει με το σύνολο των ιστοσελίδων

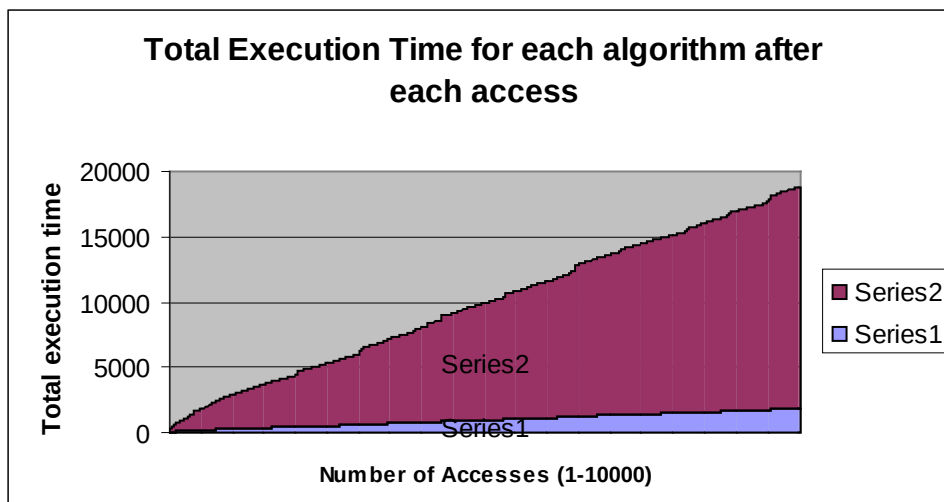
που υποτίθεται ότι είναι οι σωστές δηλαδή αυτές που πρότεινε η λύση με τους πίνακες. Τα αποτελέσματα είναι εξίσου θετικά στις περιπτώσεις που προτείνονται περισσότερες από μία ιστοσελίδες, όταν πραγματοποιείται μια έξαρση επισκέψεων. Αυτό σημαίνει ότι είναι δυνατόν να αλλάξει ο αριθμός των ανώτερων επιπέδων του δυαδικού σωρού που επιστρέφεται, ανάλογα με τον αριθμό των ιστοσελίδων που επιστρέφονται από τον ταξινομημένο πίνακα. Ακόμη και σε αυτές τις περιπτώσεις, ο αλγόριθμος αποδεικνύεται μέσω των πειραμάτων άκρως αποδοτικός και αποτελεσματικός.



Εικόνα 32 Ποσοστό Ανάκλησης με βάση τα αποτελέσματα του προτεινόμενου αλγορίθμου

4.8.3 Αποδοτικότητα από την άποψη του χρόνου απόκρισης

Δεδομένου ότι ο αλγόριθμος λειτουργεί σωστά, επόμενο μέλημα είναι να αποδειχτούν τα πλεονεκτήματα της λύσης που προτείνεται σε σχέση με τη λύση των πινάκων. Με άλλα λόγια, έχει αποδειχτεί μέσα από τα πειράματα, ότι ο αλγόριθμος επιστρέφει τις επιθυμητές προτάσεις σε λιγότερο χρόνο από εκείνον που απαιτείται από τη λύση των πινάκων ώστε να ανταποκριθεί σε μια έξαρση επισκέψεων. Στα ακόλουθα σχήματα, παρουσιάζεται ο χρόνος απόκρισης που απαιτείται από κάθε αλγόριθμο μετά από κάθε πρόσβαση σε μια ιστοσελίδα, για το ίδιο μοτίβο των προσβάσεων στα ίδια δεδομένα. Σε αμφότερες τις περιπτώσεις, προκειμένου να μετρηθεί ο χρόνος που απαιτείται για την εκτέλεση των αλγορίθμων, χρησιμοποιείται η `nanoTime()` μέθοδος της JAVA.



Εικόνα 33 Συνολικός χρόνος εκτέλεσης σε χιλιοστά του δευτερολέπτου για κάθε αλγόριθμο μετά από κάθε πρόσβαση

Στο παραπάνω σχήμα, παρουσιάζεται ο συνολικός χρόνος που έχει παρέλθει από την πρώτη πρόσβαση που γίνεται από έναν τυχαίο χρήστη μέχρι την τελευταία πρόσβαση. Με άλλα λόγια, μετά από κάθε πρόσβαση μετρίεται ο χρόνος εκτέλεσης και προστίθεται στον συνολικό χρόνο που έχει απαιτηθεί για να εξυπηρετηθεί η ακολουθία των προσβάσεων μέχρι στιγμής.

Τα αποτελέσματα που παρουσιάζονται στις εικόνες επαληθεύουν το πλεονέκτημα της προτεινόμενης λύσης, όπως αποδείχτηκε θεωρητικά, όσον αφορά την πολυπλοκότητα χρόνου. Ειδικότερα, ο συνολικός χρόνος για να ανταποκριθεί η προτεινόμενη λύση μετά από κάθε πρόσβαση είναι κατά περιπτώσεις μέχρι και 85% καλύτερος. Αυτή η διαφορά μεταξύ του χρόνου που απαιτείται από τις δύο λύσεις, μπορεί να εξηγηθεί από τη σύγκριση των θεωρητικών πολυπλοκοτήτων χρόνου. Υπενθυμίζεται ότι η χρονική πολυπλοκότητα για την προτεινόμενη λύση είναι $k \cdot \text{Log}P$ ενώ οι πίνακες απαιτούν $W \cdot \log W + P \cdot \text{Log}P + k \cdot N$, όπου W ο αριθμός των ιστοσελίδων, P ο αριθμός των κατηγοριών και N ο αριθμός των χρηστών.

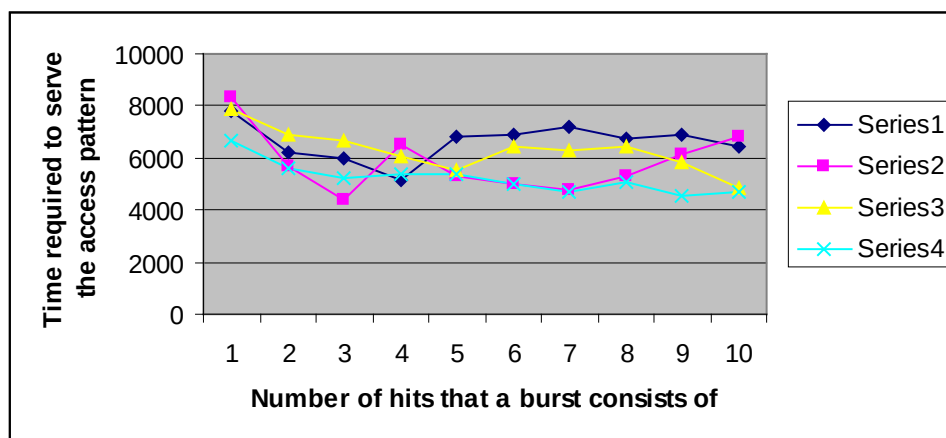
4.8.4 Αποδοτικότητα ως προς το Χρόνο με τη Χρήση Πολλαπλών Μοτίβων Εξάρσεων και Διαφορετικό Αριθμό Χρηστών και Ιστοσελίδων

Ο χρόνος που απαιτείται από την προτεινόμενη λύση προκειμένου να προτείνει μια ιστοσελίδα μπορεί να διαφέρει από εκτέλεση σε εκτέλεση, ανάλογα με τον αριθμό των προσβάσεων που μπορεί να θεωρηθεί ως μια έξαρση επισκέψεων σε μία συγκεκριμένη ιστοσελίδα, τον συνολικό αριθμό των ιστοσελίδων και τον αριθμό των χρηστών. Στην πρώτη περίπτωση, ο χρόνος εκτέλεσης του αλγορίθμου για μια έξαρση επισκέψεων που

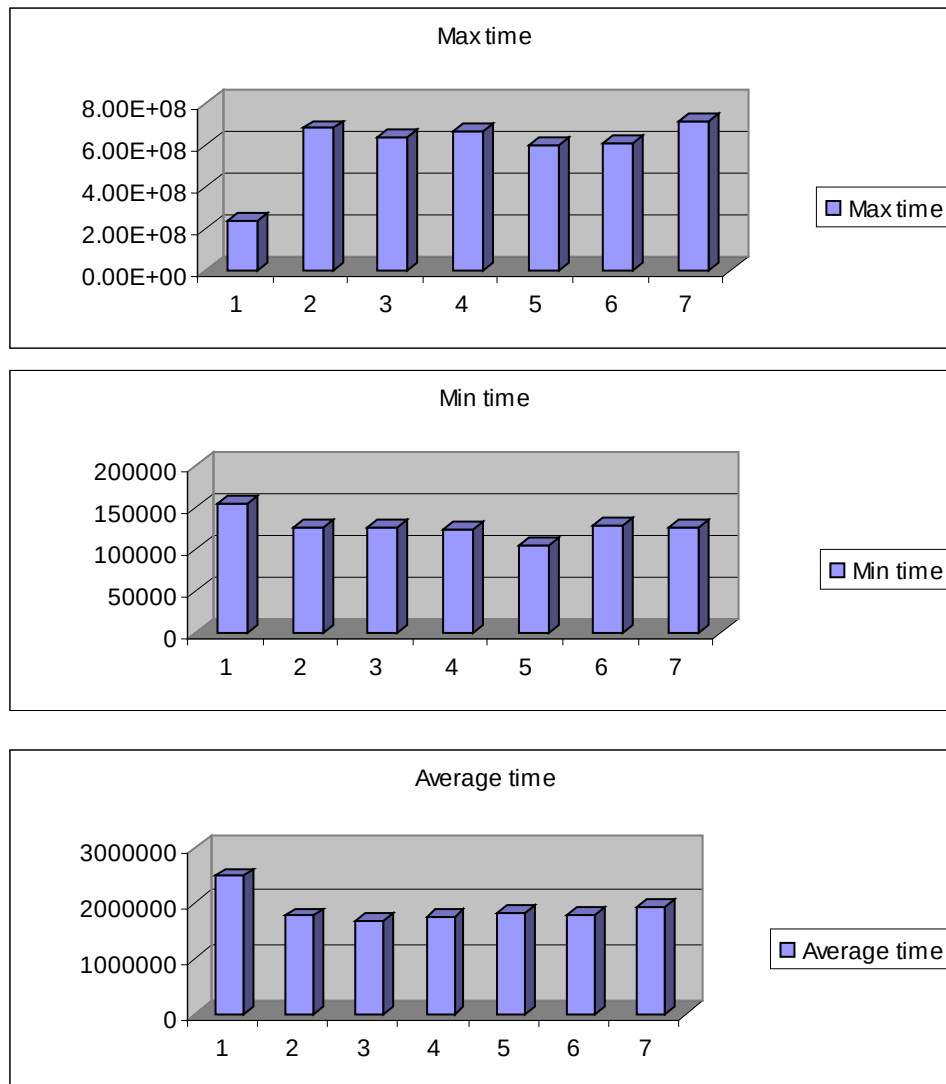
αποτελείται από 2 προσβάσεις μπορεί να είναι πολύ μεγαλύτερος ή μικρότερος από εκείνον για μια έξαρση επισκέψεων που αποτελείται από 10 προσβάσεις. Ο τελικός συνολικός χρόνος που απαιτείται, προκύπτει από τον αριθμό των εξάρσεων επισκέψεων που υπάρχουν σε μια ακολουθία προσβάσεων. Για παράδειγμα, στην περίπτωση έξαρσης επισκέψεων που αποτελείται από 2 προσβάσεις, ο χρόνος που υπολογίζεται για την εκτέλεση της ακολουθίας πρόσβασεων είναι μεγαλύτερος εάν υπάρχουν πολλές εξάρσεις σε διαφορετικές ιστοσελίδες. Αυτό σημαίνει ότι οι splay πράξεις που ακολουθούν μια έξαρση επισκέψεων είναι περισσότερες σε αυτή την περίπτωση και ως εκ τούτου αυξάνουν την τελική συνολική πολυπλοκότητα χρόνου. Αντίθετα, εάν μια έξαρση αποτελείται από 10 προσβάσεις, οι εξάρσεις που εντοπίζονται στην ακολουθία των προσβάσεων είναι λιγότερες και συνεπώς θα απαιτηθούν λιγότερες splay πράξεις και ο χρόνος μειώνεται. Από την άλλη πλευρά, υπάρχει μία σημαντική ανταλλαγή μεταξύ του μεγέθους της έξαρσης και της απόδοσης του αλγορίθμου. Κύριο μέλημα είναι να επωφεληθεί ο αλγόριθμος από την τοπικότητα του χρόνου. Επομένως, εάν μετά την έξαρση των επισκέψεων η ιστοσελίδα δε δεχθεί άλλες επισκέψεις, ο χρόνος που καταναλώθηκε για τις πράξεις splay σπαταλήθηκε και δεν υπάρχει κέρδος από την άποψη της χρονικής πολυπλοκότητας στο μέλλον. Επιπλέον, εάν η έξαρση αποτελείται από μία μόνο πρόσβαση, τότε ο αριθμός των splay πράξεων θα είναι τόσο μεγάλος, ώστε τα πλεονεκτήματα της splay τεχνικής εξαλείφονται, δεδομένου ότι όχι μόνο δαπανάται πολύς χρόνος μετά από όλες τις προσβάσεις, αλλά και ο κόμβος στον οποίο εφαρμόζεται splay αντικαθίσταται στη ρίζα του δένδρου μετά την επόμενη πρόσβαση. Στο σχήμα 34 παρουσιάζεται ο συνολικός χρόνος που απαιτείται για την εξυπηρέτηση ορισμένων ακολουθιών πρόσβασης, αλλά για διαφορετικά μεγέθη εξάρσεων. Σε όλες τις περιπτώσεις, όπως ήταν αναμενόμενο, το μέγιστο χρονικό διάστημα που απαιτείται είναι αυτό για την έξαρση μεγέθους ένα. Ο χρόνος μειώνεται για τα μεγέθη 3 και 4 και σε ορισμένες περιπτώσεις συνεχίζει να μειώνεται, όταν σε άλλες περιπτώσεις, αυξάνεται ξανά. Στις περιπτώσεις όπου υπάρχουν τυχαίες προσβάσεις και εξάρσεις, ο χρόνος μετά το μέγεθος 5 φαίνεται να είναι σταθερός. Από την άλλη πλευρά, όταν ορισμένες εξάρσεις επισκέψεων εισάγονται ή αναμένονται στην ακολουθία προσβάσεων, τότε ο αλγόριθμος μπορεί να τροποποιηθεί, ώστε να ικανοποιηθεί το δεδομένο μοτίβο πρόσβασης. Για παράδειγμα, όταν περιμένουμε εξάρσεις μεγέθους μεγαλύτερου από 10, τότε η καλύτερη απόδοση επιτυγχάνεται όταν εφαρμόζεται splay μετά από 10 συνεχόμενες προσβάσεις σε μια ιστοσελίδα. Όσον αφορά τη λύση των πινάκων, το μέγεθος της έξαρσης δεν επηρεάζει την απόδοση του αλγορίθμου, δεδομένου ότι η ταξινόμηση του πίνακα αποθήκευσης των επισκέψεων σε κάθε κατηγορία είναι αναπόφευκτη. Πιο αναλυτικά, αφού η λύση των πινάκων δεν προσπαθεί να επωφεληθεί από την τοπικότητα του χρόνου και του χώρου των προσβάσεων, ο χρόνος που απαιτείται για την ενημέρωση των αρχείων καταγραφής του ιστότοπου μετά από κάθε πρόσβαση είναι σταθερός και μη βελτιώσιμος.

Σε γενικές γραμμές, στόχος είναι η βαθμονόμηση της διαμόρφωσης του αλγορίθμου, έτσι ώστε να επιτευχθεί η βέλτιστη απόδοση σε επίπεδα ανταπόκρισης σε χρόνο και σύστασης συνδέσμων. Η βαθμονόμηση του αλγορίθμου εξαρτάται κυρίως από το πρότυπο προσβάσεων που εξυπηρετείται κάθε φορά. Για παράδειγμα, στη σειρά 2 του σχήματος 34, παρατηρείται ότι η καλύτερη επίδοση από πλευράς χρόνου επιτυγχάνεται για μια έξαρση μεγέθους 3. Το μοτίβο προσβάσεων στην περίπτωση αυτή αποτελείται κυρίως από εξάρσεις μεγέθους 10.

Στο σχήμα 35, παρουσιάζεται ο χρόνος που απαιτείται για κάθε πρόσβαση για την περίπτωση των 10.000 ιστοσελίδων, 100 χρηστών και για μεγέθη εξάρσεων από 1 έως 7. Και σε αυτή την περίπτωση, μετά από παρατήρηση των αντίστοιχων πινάκων, παρατηρείται ότι για την περίπτωση της εξάρσης μεγέθους 3 επιτυγχάνεται ο καλύτερος μέσος χρόνος που απαιτείται για κάθε πρόσβαση. Ως εκ τούτου, πριν από την εφαρμογή της προτεινόμενης λύσης σε μια ιστοσελίδα, προτείνεται να παρατηρούνται τα πρότυπα πρόσβασης στις ιστοσελίδες και να εκτελείται ο αλγόριθμος για σκοπούς βαθμονόμησης.



Εικόνα 34 Χρόνος που απαιτείται για ορισμένες ακολουθίες πρόσβασης, αλλά για διαφορετικά μεγέθη εξάρσεων (σε χιλιοστά του δευτερολέπτου).

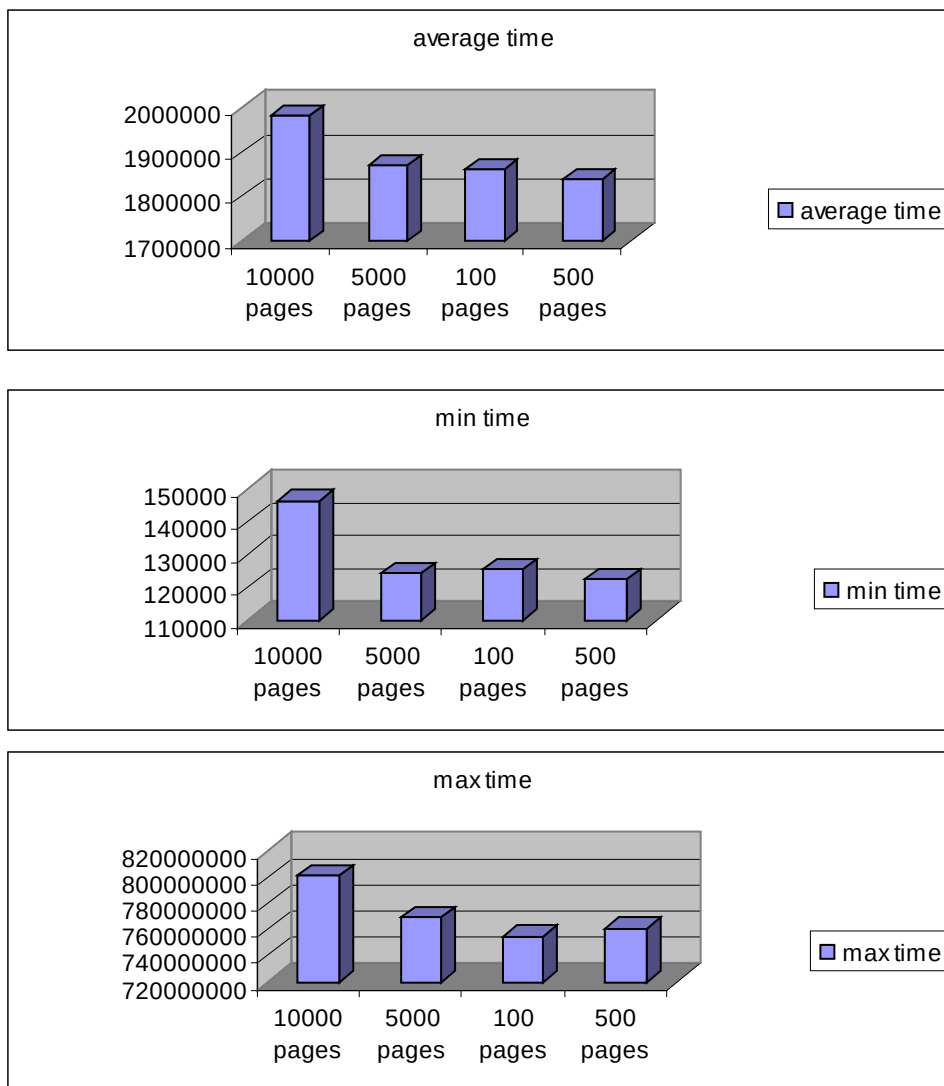


Εικόνα 35 Μέσος όρος, μέγιστος και ελάχιστος χρόνος που απαιτείται για κάθε πρόσβαση για εξάρσεις μεγεθών 1,2,3,4,5,6 και 7 επισκέψεων.

Μέγεθος Burst	1	2	3	4	5	6	7
Μέσος χρόνος	248554 8	1768764,3	1675437	1749635	1832744,4	1783887,1	1920639,7
Ελάχιστος χρόνος	155317	125057	126103	124602	105531	127823	127147
Μέγιστος χρόνος	248554 8	68042250 3	63692059 3	66477494 7	59696460 1	60524212 4	71359309 1

Πίνακας 6 Απαιτούμενος χρόνος ανά πρόσβαση για διαφορετικά μεγέθη εξάρσεων

Δεδομένου ότι η χρονική πολυπλοκότητα της προτεινόμενης λύσης εξαρτάται από τον αριθμό των χρηστών και των ιστοσελίδων, εκτελέστηκε μια σειρά πειραμάτων για να καταδειχθεί πως ο χρόνος που απαιτείται για κάθε πρόσβαση διαφέρει σε κάθε περίπτωση. Στο σχήμα 36, παρουσιάζεται ο απαιτούμενος χρόνος ανά πρόσβαση για ένα σταθερό αριθμό 100 χρηστών που εκτελούν προσβάσεις σε σύνολα ιστοσελίδων μεγέθους 500, 1000, 5000 και 10.000 ιστοσελίδων. Ο αριθμός των κατηγοριών σε κάθε περίπτωση είναι ο ίδιος. Το μέγεθος του συνόλου των ιστοσελίδων επηρεάζει το μέγεθος των splay δένδρων, δεδομένου ότι κάθε χρήστης διατηρεί ένα δέντρο με τις σελίδες που προσπελαίνει. Επομένως, μικρότερα splay δέντρα σημαίνει και φθηνότερες splay πράξεις και επομένως μια ανάλογη μείωση του χρόνου που απαιτείται για την εκτέλεση του αλγορίθμου. Τα αποτελέσματα που παρουσιάζονται στο παρακάτω σχήμα υποστηρίζουν τη θεωρία αυτή. Ειδικότερα, στην περίπτωση των 500 σελίδων, παρατηρείται ο χαμηλότερος μέσος χρόνος και αυξάνεται καθώς αυξάνεται ο αριθμός των σελίδων. Όσον αφορά το μέγιστο και ελάχιστο χρόνο, δεν οδηγούν σε ασφαλή συμπεράσματα. Αυτό είναι αναμενόμενο, δεδομένου ότι ακόμη και στην περίπτωση των 500 σελίδων, ο χρήστης μπορεί να έχει πρόσβαση σε όλες τις σελίδες και να οδηγήσουν σε μια ακριβή εκτέλεση του αλγορίθμου, ενώ στην περίπτωση των 10.000 σελίδων το μοτίβο πρόσβασης μπορεί να είναι τέτοιο, όπου κανένας χρήστης να μην έχει ένα splay δέντρο με πάνω από 500 κόμβους.

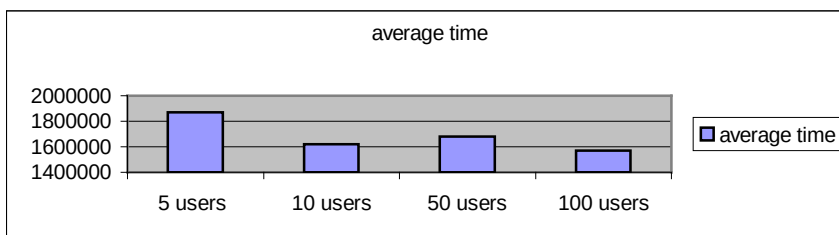


Εικόνα 36 Ο χρόνος που απαιτείται για ένα σταθερό αριθμό 100 χρηστών που εκτελούν προσβάσεις σε σύνολα ιστοσελίδων μεγέθους 500, 1000, 5000 και 10,000.

Ιστοσελίδες	10.000	5.000	1.000	500
Average	1981803,8	1868312,6	1862260,5	1837846,6
Min	146519	124266	125637	122543
Max	800941529	769727978	754604202	761032627

Πίνακας 7 Απαιτούμενος χρόνος ανά πρόσβαση για διαφορετικούς αριθμούς ιστοσελίδων

Στο επόμενο σχήμα παρουσιάζεται ο χρόνος που απαιτείται για κάθε πρόσβαση από τον αλγόριθμο στην περίπτωση των 10.000 ιστοσελίδων και πολλαπλούς αριθμούς χρηστών. Αυτό που παρατηρείται είναι ότι ο χρόνος είναι αντιστρόφως ανάλογος προς τον αριθμό των χρηστών. Η εξήγηση για αυτό το γεγονός είναι ότι όταν εξυπηρετείται το ίδιο μοτίβο προσβάσεων αλλά έχουμε λιγότερους χρήστες, τότε το splay δέντρο για κάθε χρήστη είναι σημαντικά μεγαλύτερο από ό,τι θα ήταν εάν οι προσβάσεις είχαν καταταξιωθεί μεταξύ των χρηστών. Ως εκ τούτου, μεγαλύτερα splay δέντρα σημαίνει δαπανηρές πράξεις splay και επομένως περισσότερο χρονοβόρες εκτελέσεις του αλγορίθμου. Από την άλλη πλευρά, οι δικτυακοί τόποι που η προτεινόμενη λύση μπορεί να εφαρμοστεί, είναι ιστότοποι με πολύ μεγάλη κυκλοφορία και συμπερασματικά, όσο πιο δημοφιλής είναι μία σελίδα, τόσο πιο αποδοτική και αποτελεσματική αποδεικνύεται η προτεινόμενη λύση.



Εικόνα 37 Ο χρόνος που απαιτείται για κάθε πρόσβαση στην περίπτωση των 10.000 ιστοσελίδων και πολλαπλούς αριθμούς χρηστών

	5 users	10 users	50 users	100 users
Average time	1868682	1619013	1679304	1768764
min time	150476	142732	125057	125057
Max time	39293548	45364830	6,11E+08	6,8E+08

Πίνακας 8 Απαιτούμενος χρόνος ανά πρόσβαση για διαφορετικούς αριθμούς χρηστών

4.9 Εφαρμογή σημασιολογικής έκδοσης του αλγορίθμου σε οντολογίες

Ο αλγόριθμος εξατομίκευσης που περιγράφηκε παραπάνω μπορεί να εφαρμοστεί και στο σημασιολογικό ιστό. Πιο συγκεκριμένα, με αντίστοιχο τρόπο όπως οι ιστοσελίδες οργανώνονται σε κατηγορίες, μπορούν να ανατεθούν σε οντολογίες, με βάση τη σημασιολογία τους. Η λειτουργία του αλγορίθμου ακριβώς η ίδια, απλώς πρέπει να εξηγηθεί πώς οι ιστοσελίδες μπορούν να αποδοθούν σε οντολογίες. Οντολογία στην επιστήμη των υπολογιστών είναι η εκπροσώπηση των οντοτήτων, ιδεών και εκδηλώσεων, μαζί με τις ιδιότητες και τις σχέσεις τους, σύμφωνα με ένα σύστημα οντολογιών. Χρησιμοποιείται το εργαλείο ανάλυσης ORAN Web log [24] που προσφέρει μία ολοκληρωμένη λύση ανάπτυξης και πραγματοποίησης αναλύσεων, λαμβάνοντας υπόψη τόσο τη σημασιολογία περιεχομένου του ιστότοπου αλλά και τις επισκέψεις ανά ιστοσελίδα. Μπορούν να εξαχθούν πληροφορίες σχετικά με τις προτιμήσεις του χρήστη

σε σχέση με τις θεματικές ενότητες του ιστότοπου, οι οποίες είναι πολύτιμες για το εργαλείο κατά τη διάρκεια της λήψης αποφάσεων σχετικά με τη αναδιοργάνωση του ιστότοπου. Ως εκ τούτου, πριν από την εφαρμογή του αλγορίθμου προσωποποίησης ενός ιστότοπου, χρησιμοποιείται το ORGAN εργαλείο για να εκχωρήσει τις ιστοσελίδες στις σωστές οντολογίες.

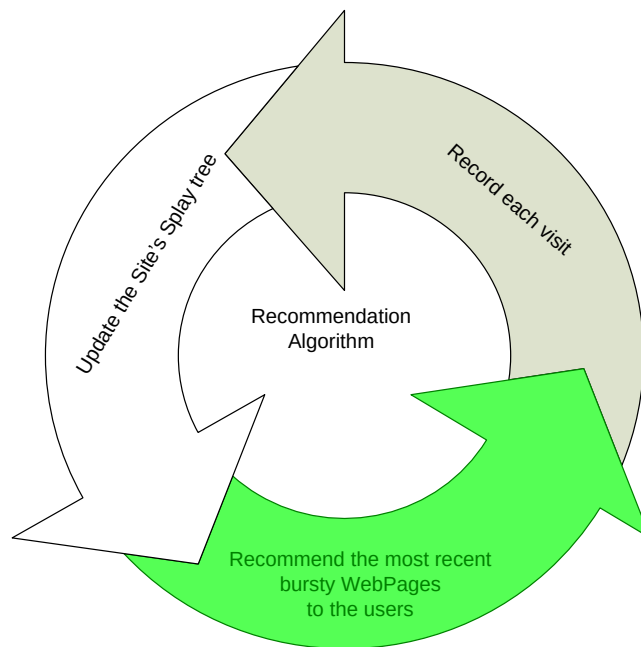
4.10 Αποδοτική Ανεύρεση Εξάρσεων Επισκέψεων

Στις επόμενες παραγράφους προτείνεται ένας αλγόριθμος ανεύρεσης εξάρσεων επισκέψεων σε ιστοσελίδες με σκοπό την αξιοποίηση της πληροφορίας σε συστήματα σύστασης αλλά και σε άλλες εφαρμογές. Καθώς αυξάνεται ο αριθμός των ιστότοπων που αποτελούνται από πολλές σελίδες, είναι πιο δύσκολο για τους επισκέπτες να φθάσουν γρήγορα στο στόχο τους. Αυτό οδηγεί σε μια επείγουσα ανάγκη για ευφυή συστήματα που υποστηρίζουν την αποτελεσματική πλοήγηση των χρηστών σε υψηλής ζήτησης περιεχόμενα στον Παγκόσμιο Ιστό. Σε πολλές περιπτώσεις, λόγω των ειδικών συνθηκών, μερικές ιστοσελίδες γίνονται πολύ δημοφιλείς και λαμβάνουν υπερβολικά μεγάλο αριθμό hits. Ως εκ τούτου, υπάρχει μεγάλη πιθανότητα ότι οι εν λόγω ιστοσελίδες να έχουν ενδιαφέρον για την πλειοψηφία των επισκεπτών σε μια δεδομένη στιγμή. Η δομή που προτείνεται μπορεί να χρησιμοποιηθεί και στην ανάπτυξη συστήματος συστάσεων (recommender system) σε ιστοσελίδες κοινωνικής δικτύωσης. Συγκεκριμένα, πιθανή εφαρμογή θα ήταν η ανεύρεση και σύσταση συνδέσμων στους οποίους παρατηρείται burst από “likes”. Τέλος, η συγκεκριμένη δομή μπορεί να χρησιμοποιηθεί στον εντοπισμό spammers και στη διαλεύκανση του “Click Fraud”. Η βασική δομή δεδομένων στην οποία βασίστηκε ο προτεινόμενος αλγόριθμος είναι και σε αυτή την περίπτωση το Splay Tree.

Τα δικτυακά συστήματα συστάσεων αποτελούν ένα συγκεκριμένο είδος τεχνικής συστήματος φιλτραρίσματος πληροφοριών που προσπαθεί να συστήσει ιστοσελίδες, που ενδέχεται να ενδιαφέρουν το χρήστη. Συνήθως, ένα σύστημα συστάσεων συγκρίνει ένα προφίλ χρήστη με κάποια χαρακτηριστικά αναφοράς, και επιδιώκει να προβλέψει την «βαθμολόγηση» που ένας χρήστης θα έδινε σε ένα στοιχείο που δεν έχει ακόμη εξεταστεί. Αυτά τα χαρακτηριστικά μπορεί να είναι από την ιστοσελίδα (προσέγγιση περιεχομένου) ή το κοινωνικό προφίλ του χρήστη (η συνεργατική προσέγγιση φιλτραρίσματος) στην περίπτωση των κοινωνικών δικτύων.

Στην συνέχεια θα περιγραφεί συνοπτικά η συνολική αρχιτεκτονική του σεναρίου της εφαρμογής του προτεινόμενου αλγορίθμου σύστασης σε ένα δικτυακό τόπο. Πρώτα από όλα, δοσμένης μιας ιστοσελίδας, οι χρήστες μπορούν να επισκεφθούν μια σειρά από ιστοσελίδες που ανήκουν σε αυτήν. Κάθε επίσκεψη σε μια ιστοσελίδα αυξάνει τη δημοτικότητα της και η επίσκεψη καταγράφεται στο αρχείο καταγραφής του δικτυακού τόπου. Ανάλογα με το μέγεθος του δικτυακού τόπου και την επισκεψιμότητά του, μπορεί να τεθούν ένα όριο επισκέψεων και ένα χρονικό διάστημα, σύμφωνα με τα οποία μπορεί να προταθεί μία επισκεπτόμενη ιστοσελίδα ως bursty. Δύο βασικά βήματα πρέπει να εκτελεστούν πριν από την επεξεργασία δεδομένων. Πρώτον, πρέπει να αναλυθεί το αρχείο καταγραφής του δικτυακού τόπου (βήμα 1), προκειμένου να συλλεχθούν πληροφορίες σχετικά με την τοπολογία / χάρτη και τη δημοτικότητα των ιστοσελίδων μέχρι στιγμής. Δεύτερον, όλες οι ιστοσελίδες πρέπει να αποθηκεύονται σε ένα γενικό

Splay δέντρο (βήμα 2) μαζί με τους μετρητές επισκέψεων τους, προκειμένου να υποβληθούν σε επεξεργασία αργότερα.



Εικόνα 38 Τα τρία κύρια στάδια του αλγορίθμου σύστασης

4.10.1 Αποδοτική Ανεύρεση Εξάρσεων Επισκέψεων

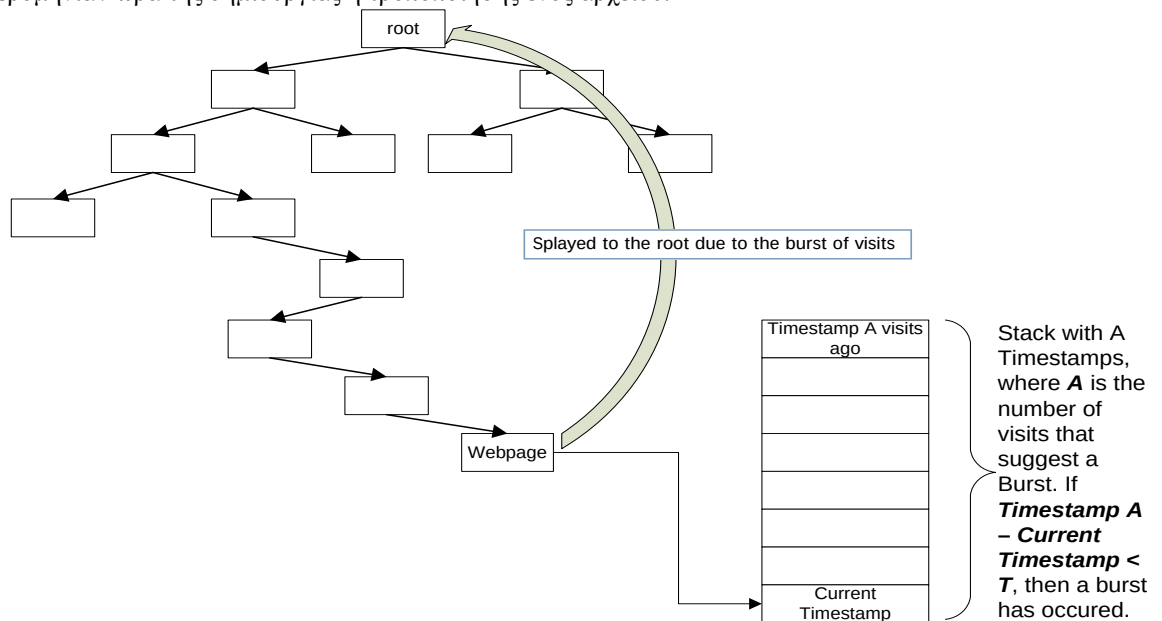
Προτείνεται ένας αλγόριθμος για την εξαγωγή των "bursty" ιστοσελίδων ενός ιστότοπου με βάση τον αριθμό των επισκέψεων της κάθε ιστοσελίδας σε ένα ορισμένο χρονικό διάστημα. Η δομή των δεδομένων που χρησιμοποιείται για τους σκοπούς του αλγορίθμου είναι το Splay δέντρο. Για να εξαχθούν οι "bursty" ιστοσελίδες αξιοποιείται η τοπικότητα χρόνου και χώρου των Splay δέντρων. Όπως αναφέραμε και πριν, μια ακολουθία των προσβάσεων σε μια ιστοσελίδα θεωρείται μια έξαρση ανάλογα με τον αριθμό των προσβάσεων (A) και το χρονικό διάστημα (T) στο οποίο έχουν πραγματοποιηθεί. Το Splay δέντρο τροποποιείται, προκειμένου να εφαρμοστεί Splay σε έναν κόμβο υπό δύο προϋποθέσεις:

- Ο κόμβος έχει προσπελαστεί A ή περισσότερες φορές

- Αυτές οι A προσβάσεις έχουν λάβει χώρα σε μια χρονική περίοδο ίση ή μικρότερη από T .

Με σκοπό την εξασφάλιση και των δύο συνθηκών γίνεται χρήση χρονοσφραγίδων (timestamps). Ένα timestamp είναι μια ακολουθία χαρακτήρων, που δηλώνει την ώρα κατά την οποία ένα συγκεκριμένο συμβάν έλαβε χώρα. Επιπλέον, ένα timestamp είναι η στιγμή κατά την οποία ένα συμβάν καταγράφεται από έναν υπολογιστή, και όχι η ώρα του ίδιου του συμβάντος. Τα στοιχεία αυτά συνήθως παρουσιάζονται σε μια συνεπή μορφή, επιτρέποντας την εύκολη σύγκριση δύο διαφορετικών καταγραφών και την παρακολούθηση της προόδου με την πάροδο του χρόνου. Η πρακτική της καταγραφής timestamps με συνεκτικό τρόπο μαζί με τα πραγματικά στοιχεία ονομάζεται Timestamping.

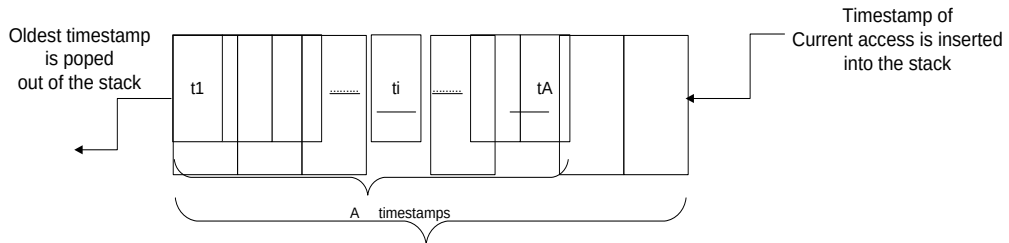
Τα timestamps συνήθως χρησιμοποιούνται για την καταγραφή γεγονότων, στην οποία περίπτωση κάθε συμβάν σε ένα αρχείο καταγραφής σημειώνεται με μια χρονική σήμανση. Στα συστήματα αρχείων, ένα timestamp μπορεί να σημαίνει την αποθηκευμένη ημερομηνία / ώρα της δημιουργίας ή τροποποίησης ενός αρχείου.



Εικόνα 39 Εφαρμόζοντας Splaying στην πιο πρόσφατη bursty ιστοσελίδα προς τη ρίζα του splay δέντρου

Στην υπό μελέτη περίπτωση, διατηρείται για κάθε κόμβο μια στοίβα μεγέθους A . Σε αυτή την στοίβα αποθηκεύονται τα timestamps των προσβάσεων που εκτελούνται στον κόμβο σύμφωνα με τη λογική FIFO. Η στοίβα είναι μεγέθους A , προκειμένου να καταγράφει το χρονικό διάστημα στο οποίο οι A προσβάσεις εκτελούνται σε έναν κόμβο και επομένως να ελέγξει την πρώτη προϋπόθεση της εκδήλωσης burst. Κάθε φορά που ένας κόμβος προσπελάζεται, το τρέχον timestamp ωθείται στη σωρό και το παλαιότερο

timestamp πετιέται έξω από τη στοίβα. Για να γίνει ο έλεγχος αν πληρείται η δεύτερη προϋπόθεση, τότε αφαιρείται το παλαιότερο timestamp από το τρέχον timestamp. Αν το αποτέλεσμα είναι ίσο ή μικρότερο από T , τότε έχει συμβεί ένα burst και ο κόμβος είναι έρχεται στη ρίζα του δέντρου μέσω της πράξης splay.



Ε ι κ ό ν α 40 Τα timestamps χρησιμοποιούνται για να διασφαλιστεί ότι ο αριθμός των προσβάσεων και το χρονικό διάστημα στο οποίο λαμβάνουν χώρα συνιστούν μια έξαρση επισκέψεων

4.10.2 Ανάλυση Χώρου και Χρόνου

- Απαιτήσεις χώρου

Όσον αφορά την πολυπλοκότητα χώρου, ο χώρος, όπως αναμενόταν, κυρίως καταναλώνεται από το Splay δέντρο και τις στοίβες των timestamps.

- Splay δέντρο: Χρειάζεται ένα Splay δέντρο για τον ιστότοπο. Στο Splay δέντρο αποθηκεύονται όλες οι ιστοσελίδες του ιστότοπου. Ως εκ τούτου, ο απαιτούμενος χώρος είναι N , όπου N είναι ο αριθμός των σελίδων του δικτυακού τόπου.

- Timestamps: Κάθε στοίβα έχει στην χειρότερη περίπτωση μέγεθος A , όπου A είναι ο αριθμός των που υποδηλώνουν μια έξαρση. Απαιτείται μία στοίβα ανά κόμβο του Splay δένδρου.

Έτσι, συνολικά, ο χώρος που απαιτείται είναι $N * A$.

- Απαιτήσεις χρόνου:

Όσον αφορά τη χρονική πολυπλοκότητα, ανά πρόσβαση απαιτούνται:

- $\log\left(\frac{\tilde{N}}{\tilde{n}}\right)$ για να εφαρμοστεί splay σε έναν κόμβο. Αυτό είναι το πολύ

$\log(\# \text{ιστοσελίδες})$.

- Χρειάζεται $O(1)$ χρόνος για να ενημερωθεί η στοίβα πρόσβασης του κόμβου και να εκτελεστεί η απαραίτητη αφαίρεση. Χρειάζεται επίσης $O(1)$ χρόνος για να επιστραφεί η ρίζα του splay δέντρου σε περίπτωση burst από επισκέψεις. Ως εκ

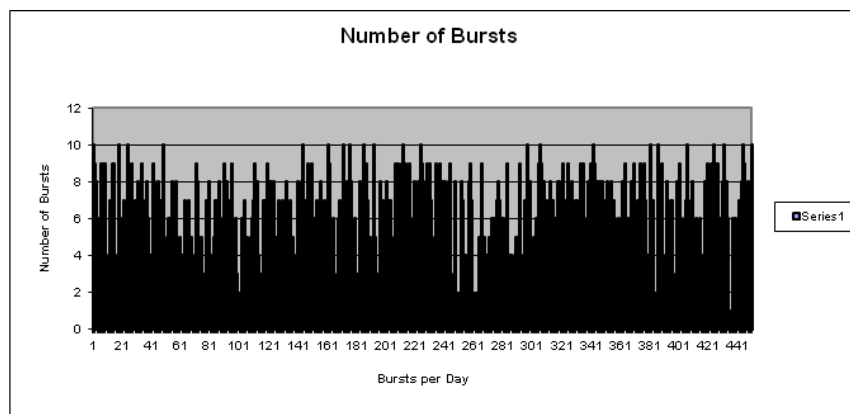
τούτου, $\log(\#web\ pages) * O(1)$ χρόνος απαιτείται για να συστήσει ο αλγόριθμος μια ιστοσελίδα μετά από την πρόσβαση σε αυτή.

4.10.3 Περιγραφή των Πειραμάτων

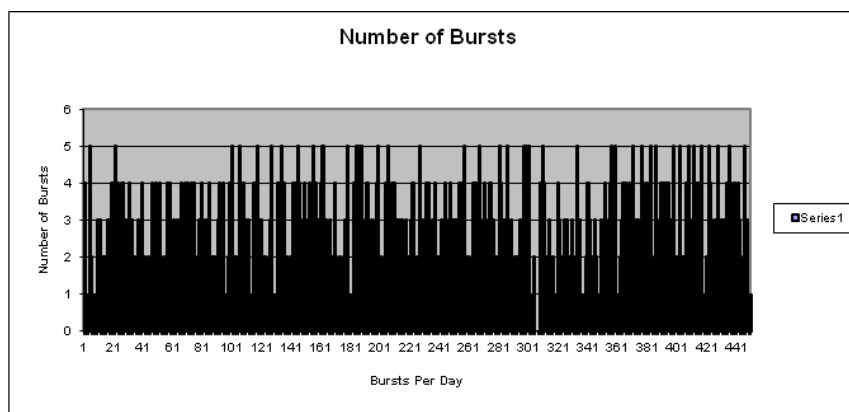
Η πειραματική διάταξη εκμεταλλεύτηκε τις λειτουργίες ενός τυπικού πραγματικού δικτυακού τόπου, με φόρουμ, fan club και ενημερωτικές σελίδες. Ο δικτυακός τόπος εμπλουτίστηκε με μια διασύνδεση των καταγεγραμμένων επισκέψεων με την εφαρμογή του αλγορίθμου σύστασης. Όλες οι καταγεγραμμένες επισκέψεις διατρέχονται από μία .NET XML Web Service που υλοποιεί τους προτεινόμενους αλγόριθμους με σκοπό να εκτελέσει μια σειρά ενεργειών που απαιτούνται:

1. Προσαρμοστική αρχικοποίηση δομών δεδομένων
2. Προσαρμοστική συντήρηση δομών δεδομένων
3. Μηχανισμός ανάδρασης επισκέψεων
4. Ανίχνευση έξαρσης
5. Σύσταση συνδέσμου
6. Πρόταση συνδέσμου και
7. Δυναμικός html μετασχηματισμός ιστότοπου - παρεμβάσεις για κάθε περίπτωση “bursty” επισκέψεων.

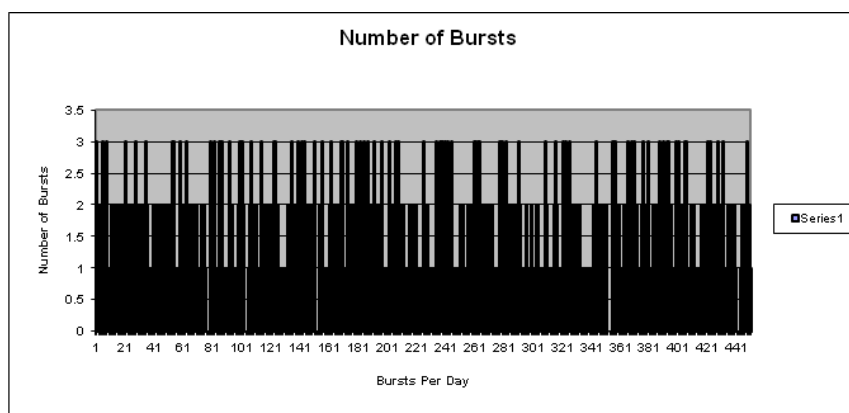
Μια σειρά διαφορετικών σεναρίων ανίχνευσης εξάρσεων εισήχθει προκειμένου να μεγιστοποιηθεί ο αριθμός των χρηστών που λαμβάνουν μια σύσταση συνδέσμου. Οι εικόνες 13-16 απεικονίζουν τη συχνότητα των εξάρσεων χρησιμοποιώντας διαφορετικά κατώτατα όρια [1000 - 4000 hits ανά χρονικό πλαίσιο].



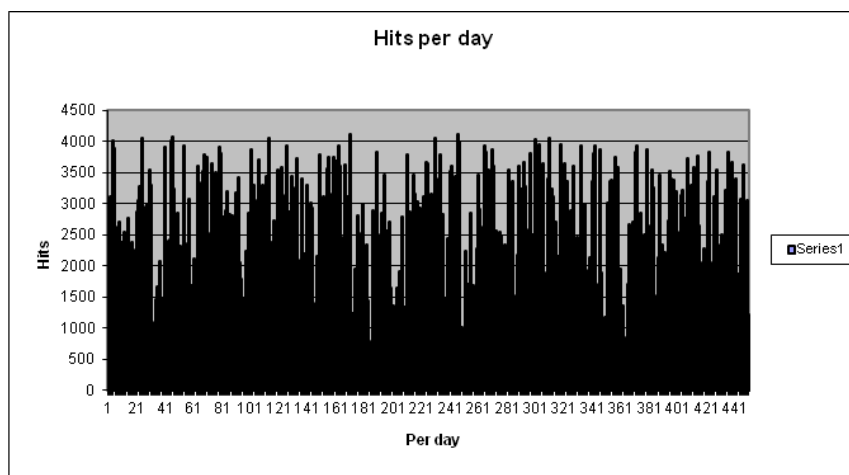
Εικόνα 41 Αριθμός Bursts για μέγεθος έξαρσης 4000 hits για το χρονικό διάστημα μιας ημέρας



Εικόνα 42 Αριθμός Bursts για μέγεθος έξαρσης 2000 hits για το χρονικό διάστημα μιας ημέρας



Εικόνα 43 Αριθμός Bursts για μέγεθος έξαρσης 1000 hits για το χρονικό διάστημα μιας ημέρας



Εικόνα 44 Εμφανίσεις ανά ημέρα για την πιο δημοφιλή σελίδα

Χρησιμοποιώντας το όριο των 4000 εμφανίσεων (hits) στην υπό μελέτη ιστοσελίδα, οι εξάρσεις που ανιχνεύονται είναι μέχρι και 15 μέσα σε μία ημέρα. Για τους λόγους της επεξεργασίας του αριθμού των επισκέψεων, προκειμένου να προσδιοριστεί ο αριθμός των εξάρσεων ανά ώρα, ο πίνακας 18 δείχνει ότι για ένα όριο 1000 hits κατά μέσο όρο, ανιχνεύονται 15 εξάρσεις ανά ώρα κατά μέσο όρο.

Οι κατωτέρω διαπιστώσεις δείχνουν ότι υπάρχουν πράγματι δύο διαφορετικές πολιτικές συστάσης που μπορούν να ακολουθηθούν. Ειδικότερα, στην περίπτωση μας, ο βασικός στόχος είναι η μεγιστοποίηση του αριθμού των χρηστών που δέχονται ένα συνιστώμενο σύνδεσμο και όχι η μεγιστοποίηση των ίδιων των συνιστώμενων συνδέσμων. Η επιλογή ποιας πολιτικής θα ακολουθηθεί είναι μια εργασία που εξαρτάται από τον αριθμό των διαθέσιμων διαδικτυακών στοιχείων στον ιστότοπο. Τα διαδικτυακά στοιχεία είναι όλα τα αντικείμενα του ιστότοπου προς τα οποία μια σύνδεση μπορεί να κατευθύνονται (δηλαδή μήνυμα στο φόρουμ, θέμα του φόρουμ, σελίδα, pdf κλπ αρχείο).

Size of Burst	1000	2000	4000
Number of Bursts	399	269	15

Πίνακας 9 Αριθμός ελάρσεων για την πιο δημοφιλή σελίδα για διαφορετικά μεγέθη εξάρσεων και το χρονικό διάστημα των 24 ωρών

Size of Burst	1000	2000	4000
Number of Bursts	15	9	2

Πίνακας 10 Αριθμός ελάρσεων για την πιο δημοφιλή σελίδα για διαφορετικά μεγέθη εξάρσεων και το χρονικό διάστημα μίας ώρας

Ο αριθμός των αντικειμένων στην ιστοσελίδα της πειραματικής διάταξης είναι μόνο μερικές χιλιάδες. Κατά συνέπεια, η πολιτική που ταιριάζει καλύτερα στην προκειμένη

περίπτωση, είναι η δεύτερη και μια σειρά από διαφορετικά μεγέθη εξάρσεων έχουν αξιολογηθεί. Το μέγεθος των εξάρσεων στη δεύτερη περίπτωση του πίνακα 10 (2000 hits ανά ώρα) αποδεικνύεται πως αποδίδει το μέγιστο αριθμό των χρηστών που λαμβάνουν μια συνιστώμενη σελίδα. Ένα παράδειγμα των συστάσεων που δίνονται μέσα σε μια ημέρα είναι το παρακάτω.

<i>Hour</i>	<i>Recommended page</i>
01:00	Idle
02:00	http://4x4fun.gr/progekset.html
03:00	http://4x4fun.gr/progekset.html
04:00	http://4x4fun.gr/progekset.html
05:00	http://4x4fun.gr/cislab/ images//programma_seminario_ian_09.jpg
06:00	http://4x4fun.gr/cislab/ images//programma_seminario_ian_09.jpg
07:00	http://4x4fun.gr/cislab/ images//programma_seminario_ian_09.jpg
08:00	http://4x4fun.gr/cislab/ images//programma_seminario_ian_09.jpg
09:00	http://4x4fun.gr/cislab/ images//programma_seminario_ian_09.jpg
10:00	http://4x4fun.gr/cislab/ images//programma_seminario_ian_09.jpg
11:00	http://4x4fun.gr/cislab/ images//programma_seminario_ian_09.jpg
12:00	http://4x4fun.gr/cislab/ images//programma_seminario_ian_09.jpg
13:00	http://4x4fun.gr/html/ prog1.pdf
14:00	http://4x4fun.gr/html/ prog1.pdf
15:00	http://4x4fun.gr/html/ prog1.pdf
16:00	http://4x4fun.gr/html/ prog1.pdf
17:00	http://4x4fun.gr/html/ prog1.pdf
18:00	http://4x4fun.gr/html/ prog1.pdf
19:00	http://4x4fun.gr/html/ prog1.pdf
20:00	http://4x4fun.gr/html/ prog1.pdf
21:00	http://4x4fun.gr/education.html
22:00	http://4x4fun.gr/education.html
23:00	http://4x4fun.gr/education.html

24:00

<http://4x4fun.gr/education.html>

**Πίνακας 11 Παράδειγμα εκτέλεσης του αλγορίθμου για μία ημέρα και για μέγεθος
έξαρσης 2000 επισκέψεις.**

Βιβλιογραφία Κεφαλαίου

1. Xin Zhang , Dennis Shasha, Better Burst Detection, Proceedings of the 22nd International Conference on Data Engineering, p.146, April 03-07, 2006
2. Zhou, B., Hui, S.C., and Chang, K., An Intelligent Recommender System Using Sequential Web Access Patterns, IEEE Conference on Cybernetics and Intelligent Systems. IEEE, Singapore, pages 393—398, 2004
3. Cho, Y., Kim, J., and Kim, S. (2002) A personalized recommender system based on web usage mining and 20th Computer Science Seminar SF2-T2-4 decision tree induction. Expert Systems with Applications. 23, 329-342.
4. VanderMeer, D., Dutta, K., and Datta, A. (2000) Enabling Scalable Online Personalization on the Web. In: Proceedings of EC'00, October 17-20, 2000, Minneapolis, Minnesota. 185-196.
5. Yu, P. (1999) Data Mining and Personalization Technologies. In: Proceedings of the International Conference on Database Systems for Advanced Applications (DASFAA99). 1-22.20th Computer Science Seminar SF2-T2-5
6. Fong, J., Hughes, J., and Zhu, J. (2000) Online Web Mining Transactions Association Rules using Frame Metadata Model. IEEE. 121-129.
7. Garrigós, I., Gómez, J., Cachero, C. (2003) Modelling Dynamic Personalization in Web Applications. ICWE 2003. 472-475.
8. Kiyomitsu, H., Takeuchi, A., Tanaka, K. (2001) Web Reconfiguration by Spatio-Temporal Page Personalization Rules Based on Access Histories. In: Proceedings of the Symposium on Applications and the Internet (SAINT 2001). IEEE Press. 75-82.
9. Magdalini Eirinaki, Michalis Vazirgiannis: Web mining for web personalization. ACM Trans. Internet Techn. 3(1): 1-27 (2003)
10. Hu, S. (2002) Helping Online Customers Decide through Web Personalization. IEEE. 17, 34-43.
11. VanderMeer, D., Dutta, K., and Datta, A. (2000) Enabling Scalable Online Personalization on the Web. In: Proceedings of EC'00, October 17-20, 2000, Minneapolis, Minnesota. 185-196.
12. Magdalini Eirinaki, Michalis Vazirgiannis: Web site personalization based on link analysis and navigational patterns. ACM Trans. Internet Techn. 7(4): (2007)
13. Antonio Picariello, Carlo Sansone A web usage mining algorithm for web personalization Source Intelligent Decision Technologies archive Volume 2 , Issue 4 (October 2008) Pages: 219-230 Year of Publication:2008
14. G. Castellano, A.M. Fanelli and M. A. Torsello (2008), “Computational Intelligence Techniques for Web personalization”. Web Intelligence and Agent Systems: an International Journal, VOL. 6, Number 3/2008, pages 253-272. Publisher: IOS Press, Amsterdam, The Netherlands. ISSN 1570-1263
15. Ranieri Baraglia, Fabrizio Silvestri: Dynamic personalization of web sites without user intervention. Commun. ACM 50(2): 63-67 (2007)

16. Jason Deane , Praveen Pathak, Ontological analysis of web surf history to maximize the click-through probability of web advertisements, *Decision Support Systems*, v.47 n.4, p.364-373, November, 2009
17. David Jingjun Xu , Stephen Shaoyi Liao , Qiudan Li, Combining empirical experimentation and modeling techniques: A design research approach for personalized mobile advertising applications, *Decision Support Systems*, v.44 n.3, p.710-724, February, 2008
18. Worapat Paireekreng, Kowit Rapeepisarn, Kok Wai Wong: Time-Based Personalised Mobile Game Downloading, *T. Edutainment 2*: 59-69 (2009)
19. Xin Zhang , Dennis Shasha, Better Burst Detection, *Proceedings of the 22nd International Conference on Data Engineering*, p.146, April 03-07, 2006
20. D.D. Sleator and R.E. Tarjan. Self-adjusting binary search trees, *Journal of the ACM*, 32:652-686, 1985.
21. Knuth, D.E. *The Art of Computer Programming*, Vol. 111 Sorting and Searching. Addison-Wesley, Reading, Mass., 1973.
22. Atkinson, M.D., J.-R. Sack, N. Santoro, and T. Strothotte (1 October 1986). "Min-max heaps and generalized priority queues.". *Programming techniques and Data structures*. Comm. ACM, 29(10): 996-1000.
23. T. Srinivasan, M. Nivedita and V. Mahadevan, 'Efficient Packet Classification using Splay Tree Models', *IJCSNS*, Vol 6, No. 5B, May 2005
24. John Garofalakis, Theodoula Giannakoudi, Evangelos Sakkopoulos , "An integrated technique for Web Site Usage Semantic Analysis: The ORGAN System", *International Workshop Logging Traces of Web Activity: The Mechanics of Data Collection in conjunction with the WWW 2006*, in the Workshop Proceedings, Edinburg, Scotland, May 23, 2006.
25. Ch. Makris, Y. Panagis, Y. Plegas, E. Sakkopoulos, An integrated web system to facilitate personalized web searching algorithms, in: *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC)*, Fortaleza, Ceara, Brazil, March 16–20, 2008, pp. 2397–2402.
26. C. Makris, Y. Panagis, E. Sakkopoulos, A. Tsakalidis, Category ranking for personalized search, *Data and Knowledge Engineering* 60 (1) (2007) 109–125.
27. D. Antoniou , E. Sakkopoulos, P. Adamopoulou, N. Tsirakis and A.Tsakalidis, "A Web Personalizing Technique using Adaptive Data Structures: the case of bursts in web visits", *Journal of Systems and Software*, Volume 83, Issue 11, November 2010, Pages 2200-2210.
28. D. Antoniou, J. Garofalakis, C. Makris, Y. Panagis and E. Sakkopoulos, "Context-similarity based hotlinks assignment: Model, metrics and algorithm", *Data & Knowledge Engineering (DKE)*, Elsevier, Vol. 69, No 4, April 2010, pp. 357-370.
29. Dimitris Antoniou, Mersini Paschou, Efrosini Sourla, Athanasios Tsakalidis, "A Semantic Web Personalizing Technique, The case of bursts in web visits", *Semantic Web Journal*, (Invited Paper).
30. D. Antoniou , E. Sakkopoulos, "A Web Recommendation Algorithm using Splay Trees: The case of burst of visits", *Data & Knowledge Engineering*, (έχει υποβληθεί).
31. Dimitris Antoniou, Mersini Paschou, Efrosini Sourla, Athanasios Tsakalidis, "A Semantic Web Personalizing Technique, The case of bursts in web visits", *International Workshop on Semantic Web And Reasoning for Cultural Heritage and Digital Libraries (SWARCH-DL)*, in ICSC 2010.

ΜΙΑ ΤΕΧΝΙΚΗ ΕΞΑΤΟΜΙΚΕΥΣΗΣ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΠΡΟΧΩΡΗΜΕΝΕΣ ΔΟΜΕΣ
ΔΕΔΟΜΕΝΩΝ: Η ΠΕΡΙΠΤΩΣΗ ΤΩΝ ΕΞΑΡΣΕΩΝ ΕΠΙΣΚΕΨΕΩΝ ΣΕ ΙΣΤΟΣΕΛΙΔΕΣ



ΑΠΟΤΕΛΕΣΜΑΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΑΝΑΚΤΗΣΗ
ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΣΤΟ ΔΙΑΔΙΚΤΥΟ

ΜΕΡΟΣ Ε: ΑΞΙΟΛΟΓΗΣΗ ΠΟΙΟΤΗΤΑΣ ΛΟΓΙΣΜΙΚΟΥ

Κεφάλαιο 5

Εφαρμογή Συσταδοποίησης για την Συνδρομή της Αξιολόγησης της Συντηρησιμότητας του Πηγαίου Κώδικα σύμφωνα με το Πρότυπο ISO/IEC-9126

Σύνοψη

Σε αυτό το κεφάλαιο παρουσιάζεται μια μεθοδολογία που βοηθά στην εκτίμηση της συντηρησιμότητας ενός προϊόντος λογισμικού σύμφωνα με το πρότυπο ποιότητας λογισμικού ISO/IEC-9126. Το κίνητρο πίσω από αυτή τη μεθοδολογία είναι η ολοκλήρωση των ομάδων των δεδομένων που εξάγονται από τη μέτρηση μετρικών που αφορούν τον πηγαίο κώδικα, με την εμπειρογνωμοσύνη των εκτιμητών του συστήματος λογισμικού, παρέχοντάς τους τη δυνατότητα να ορίσουν ένα σύνολο χαρακτηριστικών, κατάλληλων για μια τέτοιου είδους εκτίμηση.

5.1 Εισαγωγή

Το εκτιμώμενο κόστος και η απαιτούμενη προσπάθεια, σε ό,τι αφορά τη φάση συντήρησης του λογισμικού, υπερισχύουν σε μέγεθος σε σύγκριση με τις υπόλοιπες φάσεις του κύκλου ζωής του λογισμικού. Ένας σημαντικός παράγοντας, ο οποίος μέχρι πρόσφατα έτεινε να θεωρείται ως μία μικρότερης σημασίας φάση της διαδικασίας ανάπτυξης λογισμικού, είναι η συντηρησιμότητα. Παρά το γεγονός αυτό, έχει παρατηρηθεί ότι ο παράγοντας αυτός είναι απόλυτα ικανός να ασκήσει κάποια επιρροή στα στάδια της ανάπτυξης του λογισμικού.

Η συντηρησιμότητα χαρακτηρίζεται πλήρως από τα εξής 4 στοιχεία, όπως ορίζει το πρότυπο ISO9126:

- την αναλυσιμότητα,
- την τροποποιεσιμότητα,
- τη σταθερότητα και
- την ελεγχσιμότητα.

Επακόλουθο της συνεχούς και ραγδαίας ανάπτυξης των προγραμμάτων λογισμικού αποτελεί η αδιάκοπη αύξηση των απαραίτητων προσπαθειών, όπως επίσης και των απαραίτητων πόρων, οι οποίοι αποσκοπούν στη συντήρηση των προγραμμάτων αυτών.

Η εκτίμηση της συντηρησιμότητας βασίζεται τόσο στο ίδιο το λογισμικό, όσο και στη συντήρησή του από ειδικούς. Ο παράγοντας της συντηρησιμότητας χαρακτηρίζεται από δύο γενικές υποκατηγορίες: την εσωτερική και την εξωτερική συντηρησιμότητα. Από την πλευρά της εσωτερικής προσέγγισης, η μέτρηση της συντηρησιμότητας πραγματοποιείται κάνοντας χρήση κάποιων εσωτερικών μετρικών. Απώτερο σκοπό αυτού αποτελεί μία προσέγγιση της άποψης των προγραμματιστών σχετικά με τη συντηρησιμότητα ενός προϊόντος λογισμικού. Ένα βασικό χαρακτηριστικό της εσωτερικής προσέγγισης είναι ότι καθιστά τη συλλογή των αποτελεσμάτων που εξάγονται από τις μετρήσεις πιο γρήγορη και εύκολη.

Κατά τη διάρκεια της φάσης της συντήρησης, η υποβάθμιση της δομής των συστημάτων λογισμικού αποτελεί σύνθετο φαινόμενο. Η υποβάθμιση αυτή είναι απόρροια των μεταβολών - προσθηκών που πραγματοποιούνται από την αρμόδια ομάδα συντήρησης. Σαν γενικό αποτέλεσμα αυτού, η συντήρηση του συστήματος λογισμικού γίνεται ολοένα και περισσότερο δυσχερής. Στο σημείο αυτό, ο ρόλος των μετρικών λογισμικού είναι η επιβολή ενός «φράγματος» στην υποβάθμιση του συστήματος, με σκοπό η διατήρηση του συστήματος να εξακολουθεί να πραγματοποιείται σε υψηλό επίπεδο.

5.2 Χρήση Μετρικών Λογισμικού για τη Βελτίωση της Συντηρησιμότητας

Ο βαθμός της ποιότητας, ο οποίος εκφράζεται μέσω της συντηρησιμότητας, δεν υπολογίζεται εύκολα. Παρ' όλα αυτά, είναι εφικτό να μετρηθεί με έμμεσο τρόπο, έχοντας ως βάση αποτελέσματα εσωτερικών μετρικών λογισμικού. Στο σημείο αυτό, αξίζει να σημειωθεί το γεγονός ότι η πολυπλοκότητα και η δομή ενός προγράμματος σχετίζονται άμεσα με τη συντηρησιμότητα του λογισμικού.

Η καταλληλότητα των επιλεγόμενων εσωτερικών μετρικών κρίνεται με βάση τη φύση του προϊόντος λογισμικού, όπως επίσης και από τη γλώσσα προγραμματισμού στην οποία αυτό έχει υλοποιηθεί. Ακολουθούν μερικά παραδείγματα μετρικών που συνηθίζεται να χρησιμοποιούνται για την εκτίμηση του επιπέδου της συντηρησιμότητας:

- Μετρικές του Halstead [1]
- Η κυκλωματική πολυπλοκότητα του McCabe [2]
- Μετρικές του Tsai[3]
- Γραμμές κώδικα
- Γραμμές σχολίων ανά γραμμές κώδικα
- Αριθμός εισόδων και εξόδων

Μία ακόμη μέθοδος για την εκτίμηση της συντηρησιμότητας είναι η χρήση μετρικών, οι οποίες βασίζονται στη γλώσσα προγραμματισμού που χρησιμοποιείται από τα προγράμματα για τα οποία ενδιαφερόμαστε και διαφοροποιούνται ανάλογα με τον τύπο της εκάστοτε γλώσσας προγραμματισμού. Έτσι, αν η χρησιμοποιούμενη γλώσσα ανήκει στην κατηγορία των αντικειμενοστραφών γλωσσών προγραμματισμού, οι εφαρμοζόμενες μετρικές μπορεί να αφορούν το συνολικό άθροισμα των βαρών των μεθόδων μιας κλάσης, το επίπεδο συνοχής ανάμεσα στις μεθόδους, ο αριθμός των διαφορετικών κλάσεων με τις οποίες συνδέεται μία άλλη κλάση, ο αριθμός των προγόνων μίας κλάσης ή ο συνολικός αριθμός των υποκλάσεων που ανήκουν σε μία κλάση.

Αξιοσημείωτο είναι το γεγονός ότι μερικά από τα στοιχεία, που συμβάλλουν περισσότερο στη διαμόρφωση μίας άποψης σχετικά με το επίπεδο της συντηρησιμότητας, είναι δυνατό να υπολογισθούν άμεσα από τις προαναφερθείσες μετρικές. Μερικά παραδείγματα των στοιχείων αυτών αποτελούν το μέγεθος και το πλήθος των εμφωλιασμένων επιπέδων, ο αριθμός εισόδων και εξόδων μίας ρουτίνας, κλπ. Άλλωστε, είναι γνωστό ότι οι μετρικές αυτές αποσκοπούν στην εκτίμηση τόσο της δομής, της συνοχής και της σύνδεσης ανάμεσα στα διάφορα τμήματα ενός προγράμματος, όσο και της αναγνωσιμότητας, της σαφήνειας και της απλότητας του κώδικα και του επαρκούς αριθμού σχολίων στον κώδικα αυτό.

Παρ' όλα αυτά, η χρήση των εσωτερικών μετρικών είναι αποτελεσματική στις περιπτώσεις που αυτές εφαρμόζονται σε μεγάλα τμήματα κώδικα. Πιο συγκεκριμένα, έχει παρατηρηθεί ότι μετρικές που περιορίζονται σε μικρά τμήματα κώδικα, δεν αποφέρουν σαφή και συνεπώς χρήσιμα αποτελέσματα. Εξάιρεση αποτελεί η χρήση ορισμένων μετρικών, με τη βοήθεια των οποίων είναι δυνατός ο εντοπισμός τμημάτων ή ρουτινών που χαρακτηρίζονται από χαμηλή συντηρησιμότητα. Σε ό,τι αφορά τους προγραμματιστές, με τη χρήση των μετρικών, είναι σε θέση να ελέγχουν τον κώδικά τους με σκοπό τη διόρθωση και βελτίωσή του, κατά τη φάση της υλοποίησης. Η καθυστέρηση που ενδέχεται να προκύψει από τον έλεγχο αυτό που επιτελούν οι προγραμματιστές κατά τη διάρκεια της υλοποίησης είναι σαφώς προτιμότερη από την καθυστέρηση και το μεγάλο όγκο προσπαθειών και πόρων που θα προέκυπτε από τη φάση της συντήρησης του λογισμικού. Συμπερασματικά, η απόσπαση σημαντικών αποτελεσμάτων χάρη στις εσωτερικές μετρικές λογισμικού κρίνεται βασική απαίτηση για την επίτευξη επιτυχούς ολοκλήρωσης της υλοποίησης ενός προγράμματος. Έτσι, είναι εύκολο να καταλήξει κανείς στο γεγονός ότι οι μετρικές λογισμικού δίνουν αναμφισβήτητα τη δυνατότητα ενός εύκολου, αυτόματου και οικονομικού τρόπου εντοπισμού και διόρθωσης κώδικα χαμηλής ποιότητας.

Σύμφωνα με τις περιπτωσιολογικές μελέτες (case studies), εάν ένα πρόγραμμα παρουσιάζει υψηλές εσωτερικές μετρήσεις, αναμένεται να παρουσιάζει και υψηλά επίπεδα συντηρησιμότητας. Βέβαια, δεν ισχύει το ίδιο και στην αντίστροφη περίπτωση: χαμηλές εσωτερικές μετρήσεις δε συνεπάγονται χαμηλά επίπεδα συντηρησιμότητας.

Η διατήρηση των εσωτερικών μετρήσεων σε κάποια αποδεκτά όρια είναι απαραίτητη και μπορεί να πραγματοποιηθεί με τη βοήθεια ενός κατάλληλου περιβάλλοντος προγραμματισμού, το οποίο εφαρμόζει βασικές λειτουργίες εξασφάλισης ποιότητας. Ταυτόχρονα, απαιτείται η επιλογή των εσωτερικών μετρικών να είναι κατάλληλη για το αντίστοιχο πρόγραμμα προς υλοποίηση, όπως επίσης και ο καθορισμός σε αυτές των επιθυμητών φραγμάτων. Επιθυμητή είναι επίσης και η παροχή της δυνατότητας σύγκρισης διαφορετικών εκδόσεων ενός προγράμματος από τα περιβάλλοντα προγραμματισμού. Μία ακόμη δυνατότητα που προσφέρει ένα περιβάλλον προγραμματισμού είναι ο έλεγχος και η παρακολούθηση της προόδου της απόδοσης των προγραμμάτων, με βάση στόχους που έχουν τεθεί ή τις προδιαγραφές του προγράμματος ποιότητας. Με τον τρόπο αυτό, καθώς μεγαλώνει το ποσοστό των υψηλών αποτελεσμάτων που προκύπτουν από τις μετρήσεις, μειώνεται η απασχόλησή τους με τη φάση της συντήρησης. Το γεγονός αυτό έχει παρατηρηθεί ιδιαίτερα στην περίπτωση μέτρησης σε διαφορετικές εκδόσεις του ίδιου προγράμματος, καθώς οι νέες, βελτιωμένες εκδόσεις αποφέρουν υψηλότερα αποτελέσματα στις εσωτερικές μετρήσεις. Εάν σημειωθεί αποτυχία στις εσωτερικές μετρήσεις ενός τμήματος, δε θα ήταν συνετή η απόρριψή του, παρά το γεγονός ότι αυτή η αποτυχία θα καθιστούσε πιθανή και την αποτυχία στις εξωτερικές μετρήσεις. Αντίθετα, οι προγραμματιστές καλούνται να αποφανθούν, επιθεωρώντας τον κώδικα με προσοχή, εάν κάποιο τμήμα του χρειάζεται βελτίωση ή ακόμη και να γραφτεί από την αρχή, αφού υπάρχει η πιθανότητα τα χαμηλά αποτελέσματα που προέκυψαν να είναι και αναμενόμενα.

Η συντηρησιμότητα λογισμικού θεωρείται το πιο δύσκολο στάδιο στον κύκλο ζωής του λογισμικού. Σύμφωνα με το National Institute of Standards and Technology (NIST), η συντήρηση λογισμικού κοστίζει στην οικονομία των Ηνωμένων Πολιτειών της

Αμερικής \$60 δις το χρόνο [4]. Δεδομένου αυτού του μεγάλου κόστους, οι διαδικασίες συντήρησης μπορούν να θεωρηθούν ως ένα πεδίο ανταγωνιστικών πλεονεκτημάτων.

5.3 Η Προτεινόμενη Μεθοδολογία Ομαδοποίησης με Σκοπό την Αξιολόγηση της Συντηρησιμότητας Λογισμικού

Ο πρωταρχικός στόχος της προτεινόμενης μεθοδολογίας συσταδοποίησης είναι να παρέχει μια γενική αλλά διαφωτιστική άποψη ενός συστήματος λογισμικού που μπορεί να οδηγήσει τους μηχανικούς σε χρήσιμα συμπεράσματα για τη συντηρησιμότητα. Αυτή η τεχνική εξόρυξης δεδομένων είναι χρήσιμη για ανάλυση Ομοιότητας / Ανομοιότητας. Με άλλα λόγια, αναλύει ποια σημεία είναι κοντά το ένα στο άλλο σε ένα δεδομένο σύνολο δεδομένων. Με αυτό τον τρόπο, δημιουργούνται αμοιβαία αποκλειόμενες ομάδες κλάσεων σύμφωνα με τις ομοιοτήτες τους και ως εκ τούτου, η κατανόηση του συστήματος και η αξιολόγησή του διευκολύνεται. Έτσι, στους μηχανικούς συντήρησης παρέχεται μια γενική εικόνα της εξέλιξης ενός συστήματος, που τους βοηθά στην αναθεώρηση της συντηρησιμότητας του συστήματος, μελετώντας τη συμπεριφορά των κλάσεων από έκδοση σε έκδοση και ανακαλύπτοντας προγραμματιστικά μοτίβα και ασυνήθιστες ή ακραίες περιπτώσεις που μπορεί να απαιτούν μεγαλύτερη προσοχή. Προκειμένου να εξαχθούν χρήσιμες πληροφορίες για τους μηχανικούς συντήρησης μέσω της ανάλυσης συσταδοποίησης, είναι πολύ ενδιαφέρον να παρατηρείται η μορφή της κάθε ομάδας με την πάροδο του χρόνου και πώς κάθε ομάδα μεγαλώνει ή μικραίνει και πώς το μέσο σημείο της κινείται στο χώρο. Για να επιτευχθεί αυτό, προτεραιότητα είναι ο προσδιορισμός της κάθε ομάδας σε κάθε έκδοση. Μια προσέγγιση είναι να συνδυαστούν όλα τα σύνολα δεδομένων (τα σημεία δεδομένων που αντιστοιχούν στις κλάσεις) σε ένα μεγάλο σύνολο δεδομένων. Κάθε σημείο επισημαίνεται με ένα διαφορετικό χρώμα, ώστε να διαχωριστούν αργότερα. Αν εφαρμοστεί ένας αλγόριθμος συσταδοποίησης σε ένα σύνολο δεδομένων, μπορεί να γίνει η υπόθεση ότι μία ομάδα θα περιλαμβάνει στοιχεία της ίδιας ομάδας σε όλες τις εκδόσεις. Σε κάθε μία από αυτές τις ομάδες θα υπάρχουν τα ίδια στοιχεία δεδομένων με διαφορετικό χρώμα και συνεπώς από διαφορετική έκδοση. Αυτό μπορεί να επιβεβαιωθεί εφευρίσκοντας μια εσωτερική μέτρηση: το ποσοστό των σημείων δεδομένων που υπάρχουν στην ομάδα με όλους τους πιθανούς χρωματισμούς (ή ένα ποσοστό, αντιστοίχως από αυτές, όπως για παράδειγμα σε 3 από τις 5 εκδόσεις). Υπάρχουν διάφοροι τρόποι για να αξιοποιηθεί αυτή η ομαδοποίηση από αυτοματοποιημένες μεθόδους: μπορούν να ανιχνευθούν τα στοιχεία δεδομένων που έχουν ξεφύγει από την ομάδα και να εξεταστεί αν έχουν πάει σε μια καλύτερη ή χειρότερη ομάδα, εξετάζοντας σε κάθε συστάδα τις υπο-ομάδες, κάθε μία με ένα διαφορετικό χρώμα, και πώς το κεντροειδές τους κινείται και το ποσοστό των χωρικών τους επικαλύψεων. Με αυτές τις πανοραμικές παρατηρήσεις, την ακολουθία των κεντροειδών και την αναλογία της επικάλυψης, μπορεί να εξακριβωθεί αν τα στοιχεία δεδομένων της αντίστοιχης ομάδας εξελίσσονται σε καλύτερη ή σε χειρότερη κατάσταση.

Προκειμένου να προσδιοριστούν ποσοτικά οι αλλαγές στην ομάδα, ορίζεται η μετρική $m(i)$ κάθε ομάδας i , που εκφράζει πόσες παραλλαγές, στοιχεία δεδομένων (από

έκδοση σε έκδοση) υπάρχουν στην ίδια ομάδα την ίδια στιγμή, και, συνεπώς, στον ίδιο ποιοτικό χώρο. Αυτή η μετρική εκφράζεται από τον ακόλουθο τύπο:

$$m(i) = \frac{\sum_{\forall x \in i} \sum_{j=1}^n ocj(x)}{\sum_{j=1}^j pj}$$

Όπου n είναι ο αριθμός των σχηματιζόμενων ομάδων, ocj (x) είναι ο αριθμός των εμφανίσεων κάθε στοιχείου στη συστάδα i, και το pj είναι η πληθικότητα (πληθυσμός) της ομάδας i.

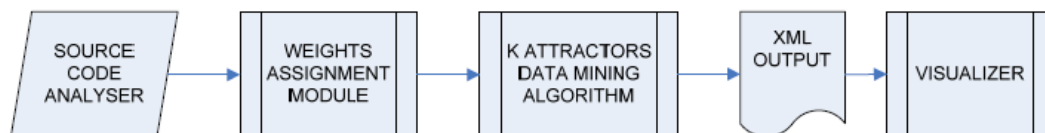
Η μεθοδολογία αναλυτικά έχει ως εξής:

1. Εξάγονται μετρικές από τον πηγαίο κώδικα του συστήματος.
2. Στις μετρικές ανατίθενται κατάλληλα βάρη, εφαρμόζοντας μια πολυκριτήρια μεθοδολογία, την Αναλυτική ιεραρχική διαδικασία, η οποία αντανακλά τη σπουδαιότητα των μετρικών στην εκτίμηση της συντηρησιμότητας. Η AHP είναι μια τεχνική λήψης αποφάσεων που επιτρέπει την εξέταση τόσο ποιοτικών όσο και ποσοτικών πτυχών στη λήψη των αποφάσεων [5]. Αναγάγει τις περίπλοκες αποφάσεις σε μια σειρά από ένα-προς-ένα συγκρίσεις και στη συνέχεια, συνθέτει τα αποτελέσματα.

Σε σύγκριση με άλλες τεχνικές, όπως η κατάταξη ή η διαβάθμιση, η AHP μιμείται την ανθρώπινη ικανότητα να συγκρίνονται οι μοναδικές ιδιότητες των εναλλακτικών λύσεων. Επιπλέον η AHP όχι μόνον βοηθά τους υπεύθυνους λήψης αποφάσεων να επιλέξουν την καλύτερη εναλλακτική λύση, αλλά επίσης παρέχει μια σαφή αιτιολόγηση της επιλογής. Με συστηματικό τρόπο η AHP συγκρίνει τη λίστα των στόχων ή τις εναλλακτικές λύσεις. Όταν χρησιμοποιείται στη διαδικασία μηχανικής συστημάτων, η AHP μπορεί να αποδειχτεί ως ένα ισχυρό εργαλείο για τη σύγκριση των εναλλακτικών σχεδίων. Έστω πως έχει καθοριστεί ένα σύνολο στόχων και ότι γίνεται προσπάθεια να θεσπιστεί μια κανονικοποιημένη σειρά των βαρών που πρέπει να χρησιμοποιούνται για τη σύγκριση των εναλλακτικών λύσεων με βάση αυτούς τους στόχους. Η AHP σχηματίζει ένα πίνακα ζεύγων συγκρίσεων A, όπου ο αριθμός που αντιστοιχεί στην i-οστή γραμμή και j-οστή στήλη δίνει τη σχετική σημασία του αντικειμένου O (i) σε σύγκριση με το O (j). Οι τιμές που συνήθως χρησιμοποιούνται είναι σε μια κλίμακα 1-9, με όπου A(i, j) = 1 εάν τα δύο αντικείμενα είναι ίσης σημασίας, A(i, j) = 3, εάν το O(i) είναι ασθενώς περισσότερο σημαντικό από το O (j), A(i, j) = 5, εάν το O (i) είναι σημαντικά πιο σημαντικό από O(j), A(i, j) = 7 εάν το O(i) είναι πολύ έντονα πιο σημαντικό από το O(j), και A(i, j) = 9, εάν το O(i) είναι απόλυτα πιο σημαντικό από το O(j). Μετά από αυτό τη διαδικασία, ο πίνακας συγκρίσεων κανονικοποιείται και υπολογίζονται οι ιδιοτιμές τους. Αυτές οι ιδιοτιμές παίζουν το ρόλο των συντελεστών / βαρών όταν κάποιος θέλει να αξιολογήσει τις εναλλακτικές λύσεις για τους υπό εξέταση στόχους. Στην περίπτωση αυτή, στόχος είναι να αξιολογηθεί η συντηρησιμότητα μέσω ενός συνόλου εφαρμοζόμενων μετρικών. Εφαρμόζεται η διαδικασία AHP σε κάθε

επίπεδο της ιεραρχίας των μετρικών συντηρησιμότητας. Στο πρώτο επίπεδο αξιολογούνται τα χαρακτηριστικά από τις εξαγόμενες μετρικές και σε δεύτερο επίπεδο γίνεται αξιολόγηση της συντηρησιμότητας από τα χαρακτηριστικά με την εφαρμογή της AHP διαδικασίας και πάλι. Έτσι, στο πρώτο επίπεδο έχει κατασκευαστεί ένας συγκριτικός πίνακας ανά ζεύγη, για κάθε ένα από τα χαρακτηριστικά που αντανakλούν τη γνώση του εμπειρογνώμονα σχετικά με το επηρεάζει κάθε μετρική κάθε χαρακτηριστικό. Στη συνέχεια, εφαρμόζοντας την κανονικοποίηση και την εξαγωγή των ιδιοτιμών, από κάθε πίνακα υπολογίζεται το βάρος κάθε μετρικής για τον υπολογισμό ενός αποτελέσματος για κάθε χαρακτηριστικό. Στο υψηλότερο επίπεδο, κατασκευάζεται ένας πίνακας σύγκρισης ανά ζεύγη, αντανakλώντας την γνώση του εμπειρογνώμονα σχετικά με το πόσο κάθε χαρακτηριστικό επηρεάζει τη συντηρησιμότητα. Τα βάρη υπολογίζονται με την κανονικοποίηση και την εξαγωγή των ιδιοτιμών. Υπάρχουν πολλές δυνατότητες για την αξιολόγηση διαφορετικών συστημάτων:

- Αξιολόγηση όλων των συστημάτων χρησιμοποιώντας τα ίδια βάρη που παράγονται από την υποκειμενικότητα ενός εμπειρογνώμονα λογισμικού.
- Αξιολόγηση όλων των συστημάτων χρησιμοποιώντας το μέσο βάρος που παράγεται από την υποκειμενικότητα πολλών εμπειρογνομένων λογισμικού.
- Αξιολόγηση του κάθε συστήματος ανεξάρτητα, αξιοποιώντας τα βάρη που παράγονται από την υποκειμενικότητα ενός εμπειρογνώμονα λογισμικού που εστιάζει στα χαρακτηριστικά του συστήματος.



Εικόνα 45 Αρχιτεκτονική Συστήματος Αξιολόγησης Συντηρησιμότητας Πηγαίου Κώδικα

3. Ο αλγόριθμος ομαδοποίησης k-Attractors [6] εφαρμόζεται στις τιμές της συντηρησιμότητας που έχουν προκύψει σύμφωνα με το ISO/IEC-9126, με σκοπό να προσφερθεί στον εκτιμητή μία γενική και γρήγορη επισκόπηση του συστήματος.

Η προτεινόμενη μεθοδολογία υποστηρίχθηκε από το εργαλείο Code4Thought [7]. Οι ενότητες που συνιστούν και υλοποιούνται από το εργαλείο αυτό είναι:

- Η εξαγωγή και προετοιμασία των δεδομένων
- Η ανάθεση βαρών
- Η ανάλυση των δεδομένων

Ανάλυση Δεδομένων

Όπως φαίνεται στην εικόνα, ο αλγόριθμος k-Attractors δέχεται σαν είσοδο τα δεδομένα που προέρχονται από τον αναλυτή του πηγαίου κώδικα, εκτελώντας ερωτήματα πάνω στη βάση δεδομένων. Το αποτέλεσμα της ανάλυσης αποθηκεύεται σε αρχεία XML, με σκοπό την οπτικοποίησή της από την κατάλληλη ενότητα. Στην περίπτωση της εκτίμησης της συντηρησιμότητας λογισμικού, η ομαδοποίηση προσφέρει επισκοπήσεις των συστημάτων δημιουργώντας αμοιβαίως αποκλειόμενες ομάδες κλάσεων, ιδιωτικών χαρακτηριστικών ή μεθόδων, σύμφωνα με τις ομοιότητές τους σε σχέση με τις μετρήσεις επάνω στον πηγαίο κώδικα. Η διαδικασία αυτή συμβάλλει στη

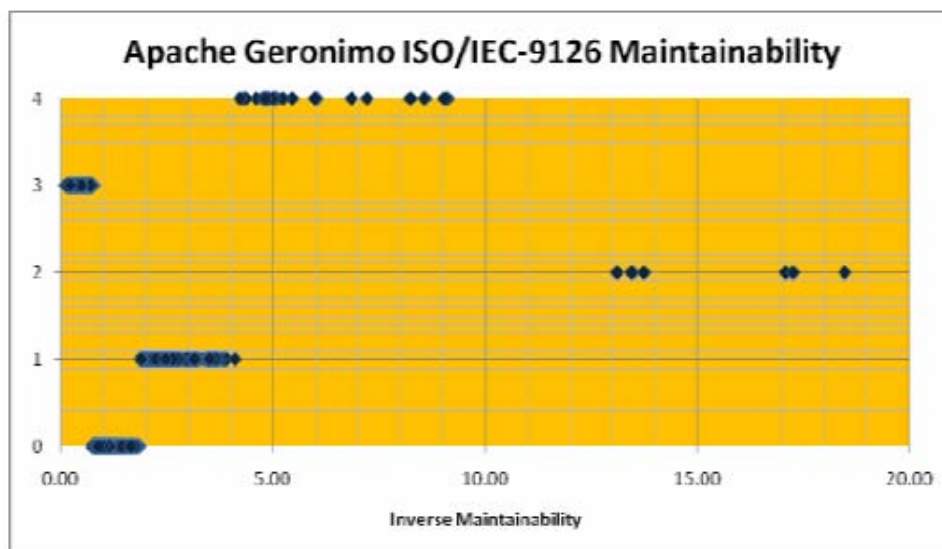
μείωση του χρόνου που απαιτείται στην κατανόηση και συνολική αξιολόγηση του συστήματος.

Μία άλλη σημαντική συμβολή της ομαδοποίησης είναι ότι βοηθά στην ανακάλυψη προγραμματιστικών μοτίβων αλλά και ακραίων τιμών, οι οποίες ενδεχομένως να χρήζουν ιδιαίτερης προσοχής σε θέματα συντηρησιμότητας. Για αυτούς τους λόγους, χρησιμοποιήθηκε ο αλγόριθμος k-Attractors προσαρμοσμένος σε αριθμητικά δεδομένα όπως οι μετρικές πηγαίου κώδικα. Τα κυριότερα χαρακτηριστικά του k- Attractors είναι ότι καθορίζει την επιλογή των αριθμών των ομάδων χωρίς την παρέμβαση του χρήστη, εντοπίζει τα αρχικά σημεία προσέλευσης των κέντρων των ομάδων με μεγάλη ακρίβεια και υπολογίζει την ομοιότητα σύμφωνα με μια σύνθετη μετρική που συνδυάζει την απόσταση Hamming και το εσωτερικό γινόμενο των συνδιαλλαγών και των κέντρων των ομάδων.

Ο αλγόριθμος k-Attractors χρησιμοποιεί την τεχνική της ανακάλυψης των μέγιστων συχνών στοιχειοσυνόλων (maximal frequent itemset) και τη τμηματοποίηση με σκοπό τον ορισμό του αριθμού των επιθυμητών ομάδων και των αρχικών σημείων προσέλευσης των κέντρων αυτών των ομάδων. Η διαίσθηση πίσω από αυτή την πολιτική, είναι ότι ένα συχνό στοιχειοσύνολο στην περίπτωση των μετρικών λογισμικού, είναι ένα σύνολο μετρήσεων που εντοπίζονται μαζί σε ένα ελάχιστο τμήμα των κλάσεων του συστήματος λογισμικού. Οι κλάσεις με παρόμοιες μετρήσεις αναμένεται να ανήκουν στην ίδια ομάδα. Ο όρος σημείο προσέλευσης προτιμάται αντί του όρου κεντροειδές, εφόσον δεν επιλέγεται τυχαία αλλά με βάση τη συχνότητα σε σχέση με το σύνολο των κλάσεων ενός συστήματος λογισμικού.

5.4 Εφαρμογές - Αξιολόγηση αποτελεσμάτων

Η προτεινόμενη τεχνική εφαρμόστηκε στον εξυπηρετητή εφαρμογών Apache's Geronimo. Η αξιολόγηση της συντηρησιμότητας Geronimo σύμφωνα με το ISO/IEC-9126, περιλαμβάνει τη μελέτη 1440 κλάσεων. Η εικόνα 46 αναπαριστά τις ομάδες που προέκυψαν από την ομαδοποίηση των τιμών των κλάσεων του Geronimo με βάση τη συντηρησιμότητα. Όσο πιο μεγάλες είναι οι τιμές στον άξονα χ , τόσο λιγότερο συντηρήσιμες είναι οι κλάσεις. Στον πίνακα 12 παρουσιάζονται τα στατιστικά των ομάδων που προέκυψαν.



Εικόνα 46 Συστάδες Συντηρησιμότητας του Apache Geronimo σύμφωνα με το ISO 9126

S/N	Population	Percentage	Mean	Standard Deviation
0	419	29%	1.10	0.29
1	130	9%	2.45	0.60
2	7	0.004%	13.75	2.27
3	856	59%	0.39	0.16
4	28	1.996%	5.02	1.55

Πίνακας 12 Στατιστικά Συστάδων

Η ομάδα, η οποία έχει το μεγαλύτερο πληθυσμό, περιέχει κλάσεις που η τιμή της συντηρησιμότητάς τους κυμαίνεται μεταξύ 0 και 0.9. Αυτό δείχνει ότι η συντριπτική πλειονότητα των κλάσεων του Geronimo είναι ιδιαίτερα συντηρήσιμες. Επιπλέον, οι ομάδες 0, 1 και 4 περιέχουν κλάσεις που η τιμή της συντηρησιμότητάς κυμαίνεται από 0,9 - 2, 2 - 4 και 4 - 9,2, αντίστοιχα, τιμές που μπορούν να θεωρηθούν καλές από την άποψη της συντηρησιμότητας. Ωστόσο, οι ακραίες τιμές ανιχνεύονται στην ομάδα 2, η οποία αποτελείται από μόνο επτά (7) κλάσεις που έχουν τις χαμηλότερες τιμές συντηρησιμότητας.

S/N	WMC	NPM	DAM	CBO	POM	DIT	NOC	LCOM	Ca
1	9.15	11.13	1.62	17.40	40.00	0.72	0.00	42.69	0.00
2	11.58	4.52	1.62	35.65	30.00	0.72	0.00	45.21	0.51
3	10.68	0.32	1.62	2.99	2.50	0.72	0.00	81.97	2.53
4	18.38	11.45	1.62	14.37	20.00	0.72	0.00	64.96	9.61
5	14.77	11.29	1.62	13.14	12.50	0.72	0.00	47.82	9.61
6	0.42	0.81	0.00	0.33	0.00	0.72	149.49	0.27	26.30
7	0.28	0.48	0.00	0.00	0.00	1.44	76.27	0.18	52.10

Πίνακας 13 Μετρικές Συστάδας 2

Ο πίνακας 13 παρουσιάζει τις τιμές των μετρικών για τις κλάσεις της ομάδας 2. Μια περαιτέρω μελέτη των τιμών αυτών δείχνουν ότι οι κλάσεις στην ομάδα 2 μπορούν να ομαδοποιηθούν σε δύο κατηγορίες:

Η πρώτη κατηγορία περιλαμβάνει τις πέντε πρώτες κλάσεις που έχουν τα ακόλουθα χαρακτηριστικά:

- Δεν ακολουθούν την αρχή της χαμηλής ζεύξης / υψηλής συνοχής. Αντίθετα εμφανίζουν χαμηλή συνοχή και υψηλή ζεύξη.
- Είναι εξαιρετικά πολύπλοκες.
- Όλες έχουν πολυμορφικές μεθόδους. Το γεγονός αυτό δείχνει ότι ενθυλάκωση δεν εφαρμόζεται σε αυτές τις κλάσεις.

Η δεύτερη κατηγορία περιλαμβάνει τις κλάσεις ASN1Encodable και DERObject που είναι δύσκολο να συντηρηθούν για διαφορετικούς λόγους. Πιο συγκεκριμένα οι δύο αυτές κλάσεις έχουν τα εξής χαρακτηριστικά:

- Είναι ενδιαφέρον ότι δεν είναι περίπλοκες, και το μέγεθός τους είναι πολύ μικρό σε αντίθεση με τις κλάσεις που αφορούν την πρώτη κατηγορία. Επίσης ακολουθούν την αρχή της χαμηλής ζεύξης / υψηλής συνοχής.
- Έχουν υπερβολικό αριθμό παιδιών. Αυτό δείχνει μάλλον ότι οι εν λόγω κλάσεις αποτελούν θεμελιώδη στοιχεία της δομής του Apache Geronimo.
- Ο αριθμός των κλάσεων που εξαρτώνται από αυτές (Ca) είναι μεγάλος.

Ο Πίνακας 12 παρουσιάζει στατιστικά στοιχεία για τις μετρικές των κλάσεων του Apache Geronimo στις ομάδες 0, 1, 3 και 4. Ο πίνακας αυτός δείχνει ότι:

- Όσο χαμηλότερες είναι οι τιμές των μετρικών τόσο μεγαλύτερη είναι η πιθανότητα χαμηλής συντηρησιμότητας.
- Υπάρχει περιορισμένη χρήση της κληρονομίας, όπως φαίνεται από τις χαμηλές τιμές των DIT και NOC.
- Η πλειοψηφία των κλάσεων ακολουθεί την αρχή της χαμηλής ζεύξης / υψηλής συνοχής.
- Οι περισσότερες κλάσεις παρουσιάζουν χαμηλή πολυπλοκότητα.
- Η σχεδιαστική ιδιότητα της ενθυλάκωσης εφαρμόζεται στις περισσότερες από τις κλάσεις.

5.5 Μελλοντικές εργασίες

Τα ευρήματά μας υποδεικνύουν ότι η προτεινόμενη μεθοδολογία παρουσιάζει εξαιρετική σημασία σε ό,τι αφορά τη διευκόλυνση των μηχανικών συντήρησης στην παρακολούθηση της εξέλιξης της συντηρησιμότητας ενός συστήματος. Από την άλλη πλευρά, στερείται τη δυνατότητα να προβλέψει τη συντηρησιμότητα μίας επερχόμενης έκδοσης του συστήματος. Μία άλλη τεχνική εξόρυξης δεδομένων, η οποία διαθέτει ικανότητα πρόβλεψης (όπως η ταξινόμηση), θα μπορούσε να εφαρμοσθεί επιπρόσθετα, έτσι ώστε να ενισχυθεί η μεθοδολογία μας. Επιπλέον και ανεξάρτητα από αυτό, θεωρούμε τις ακόλουθες εναλλακτικές, με σκοπό να αναπτύξουμε περαιτέρω την προτεινόμενη μεθοδολογία.

- Συσταδοποίηση των συνιστωσών των συστημάτων, με βάση τις δυναμικές τους εξαρτήσεις. Θα ήταν εξαιρετικά ενδιαφέρονσα η προσπάθεια να αξιολογήσουμε τη χρησιμότητα της ανάλυσης των δυναμικών εξαρτήσεων των εξαρτημάτων ενός συστήματος λογισμικού.

- Η εφαρμογή μίας εναλλακτικής προσέγγισης για την παρακολούθηση της συσταδοποίησης παρουσιάζει παραλλαγές ανάλογα με την έκδοση. Μία άλλη προσέγγιση για την παρακολούθηση των μεταβολών συσταδοποίησης αφορά την πραγματοποίηση της διαδικασίας συσταδοποίησης για κάθε έκδοση. Χρησιμοποιούμε όλες αυτές τις συστάδες (καθεμία με διαφορετικό χρώμα, ανάλογα με την έκδοση στην οποία ανήκει) σε μία δεύτερη φάση συσταδοποίησης, χρησιμοποιώντας τα αντίστοιχα κεντροειδή, με σκοπό να παράγουμε συστάδες συστάδων ταξινομημένες ιεραρχικά. Μπορούμε να συμπεράνουμε ότι κάθε συστάδα του δεύτερου επιπέδου αποτελείται από την ίδια συστάδα αντικειμένου πληροφορίας, ανάλογα με την έκδοση.

- Ενίσχυση της μεθόδου εξαγωγής. Η προτεινόμενη μέθοδος κατεργάζεται τις πληροφορίες αυτές, οι οποίες προέρχονται μόνο από αρχείο πηγαίου κώδικα Java (.java). Επίσης, εξίσου ενδιαφέρονσα είναι η εξαγωγή δεδομένων από άλλες γλώσσες, όπως η C++, C και COBOL.

Βιβλιογραφία Κεφαλαίου

1. Halstead M.H., “Elements of Software Science”, Elsevier Publications, N-Holland, 1975.
2. McCabe T.J., “A complexity measure”, IEEE Transactions in Software Engineering SE-2 (4), pp. 308-320, 1976.
3. Tsai W.T., Lopez M.A., Rodriguez V. and Volovik D., “An Approach to Measuring Data Structure Complexity”, Compsac86, pp. 240-246, 1986.
4. National Institute of Standards and Technology (NIST), “The Economic Impacts of Inadequate Infrastructure for Software Testing.”, Washington D.C. 2002.
5. Saaty T., Multicriteria Decision Making: The Analytic Hierarchy Process, Vol. 1, AHP Series, RWS Publications, 502 pp., 1990
6. Kanellopoulos Y., Antonellis P. Tjortjis C., Makris C., “k-Attractors, A Clustering Algorithm for Software Measurement Data Analysis”, In Proceedings of IEEE 19th International Conference on Tools for Artificial Intelligence (ICTAI 2007), IEEE Computer Society Press 2007
7. <http://www.code4thought.org>

8. Antonellis P., Antoniou D., Kanellopoulos Y., Makris C., Theodoridis E., Tjortjis C., Tsirakis N., "Clustering for Monitoring Software Systems Maintainability Evolution", Electronic Notes in Theoretical Computer Science, (Elsevier), Vol. 233, March 2009, pp. 43-57, ISSN 1571-0661.
9. Kanellopoulos Y., Antonellis P., Antoniou D., Makris C., Theodoridis E., Tjortjis C., Tsirakis N., "Code Quality Evaluation methodology using the ISOIEC 9126 Standard", Int'l Journal of Software Engineering & Applications, Vol.1, No.2, July 2010, pp. 17-36, (AIRCC), ISSN 0976-2221.
10. P. Antonellis, D. Antoniou, Y. Kanellopoulos, C. Makris, E. Theodoridis, C. Tjortjis, N. Tsirakis, 11th European Conference on Software Maintenance and Reengineering (CSMR), March 2007. A Data Mining Methodology for Evaluating Maintainability according to ISO/IEC-9126 Software Engineering-Product Quality Standard.
11. P. Antonellis, D. Antoniou, Y. Kanellopoulos, C. Makris, E. Theodoridis, C. Tjortjis and N. Tsirakis, "Clustering for Monitoring Software Systems Maintainability Evolution". European Conference on Software Maintenance and Reengineering (CSMR) 2008, International Workshop on Software Quality and Maintainability (SQM 2008)
12. Panos Antonellis, Dimitris Antoniou, Yiannis Kanellopoulos, Christos Makris, Christos Tjortjis, Vangelis Theodoridis, Nikos Tsirakis, "Code4Thought Project: Employing the ISO/IEC-9126 Standard for Software Engineering-Product Quality Assessment," csmr, pp.297-300, 2009 European Conference on Software Maintenance and Reengineering, 2009
13. Antonellis P., Antoniou D., Theodoridis E., Kanellopoulos Y., Makris C., Tjortjis C., Tsirakis N., "Employing Clustering for Assisting Source Code Maintainability Evaluation according to ISO/IEC-9126" Artificial Intelligence Techniques in Software Engineering Workshop (AISEW 2008) in ECAI08.



ΑΠΟΤΕΛΕΣΜΑΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΑΝΑΚΤΗΣΗ
ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΣΤΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ

Ευρετήριο Όρων

C

Chain Splay.....50

H

Hotlink70

S

Splay δέντρο43
Simetric.....76

A

Αναδιάταξη Ιστότοπου.....70

Ανεύρεση Εξάρσεων121

E

Έξαρση Επισκέψεων103,121
Εξατομίκευση Ιστότοπου96

M

Μετρικές ποιότητας λογισμικού.....135

Σ

Συντηρησιμότητα Λογισμικού134
Συμπύεση Δεδομένων55