

Υπολογισμός κρίσιμων κόμβων σε
κατευθυνόμενα γραφήματα

Η Μεταπτυχιακή Διπλωματική Εργασία

υποβάλλεται στην ορισθείσα

από τη Συνέλευση

του Τμήματος Μηχανικών Η/Υ και Πληροφορικής

Εξεταστική Επιτροπή

από τον

Φώτιο Γραμμένο

ως μέρος των υποχρεώσεων για την απόκτηση του

ΔΙΠΛΩΜΑΤΟΣ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΣΤΗ ΜΗΧΑΝΙΚΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

ΜΕ ΕΙΔΙΚΕΥΣΗ
ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΚΑΙ ΜΗΧΑΝΙΚΗ ΔΕΔΟΜΕΝΩΝ

Πανεπιστήμιο Ιωαννίνων

Πολυτεχνική Σχολή

Ιωάννινα 2024

Εξεταστική Επιτροπή:

- **Λουκάς Γεωργιάδης**, Αναπλ. Καθηγητής, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Ιωαννίνων (Σύμβουλος)
- **Χρήστος Νομικός**, Καθηγητής, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Ιωαννίνων
- **Χάρης Παπαδόπουλος**, Αναπλ. Καθηγητής, Τμήμα Μαθηματικών, Πανεπιστήμιο Ιωαννίνων

ΑΦΙΕΡΩΣΗ

Στην οικογένεια μου, για την πίστη τους σε εμένα και τις θυσίες που έκαναν ώστε να μου προσφέρουν τη δυνατότητα να φτάσω ως εδώ. Σε όλους τους φίλους, συναδέλφους και καθηγητές μου, που με στήριξαν και με ενθάρρυναν σε κάθε βήμα αυτού του ταξιδιού. Και, τέλος, αφιερώνω αυτή την εργασία στη μνήμη εκείνων που έχουν φύγει από τη ζωή, αλλά των οποίων η επιρροή και οι αξίες συνεχίζουν να με καθοδηγούν και να με εμπνέουν κάθε μέρα.

ΕΥΧΑΡΙΣΤΙΕΣ

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κ. Λουκά Γεωργιάδη, για την καθοδήγησή του, την άμεση ανταπόκριση, την αμέριστη υποστήριξη και την εμπιστοσύνη που μου έδειξε με την ανάθεση του θέματος της μεταπτυχιακής διπλωματικής εργασίας. Επιπρόσθετα, οφείλω ένα μεγάλο ευχαριστώ στους γονείς μου και την αδερφή μου για την βοήθεια και όλη την στήριξη κατά την πορεία της διπλωματικής εργασίας, καθώς και τους αγαπημένους μου φίλους.

ΠΕΡΙΕΧΟΜΕΝΑ

Κατάλογος Σχημάτων	iii
Κατάλογος Πινάκων	iv
Κατάλογος Αλγορίθμων	vi
Περίληψη	vii
Extended Abstract	viii
1 Εισαγωγή	1
1.1 Κίνητρο - Στόχοι	1
1.2 Σχετική δουλειά	4
1.3 Συνεισφορά	6
1.4 Σχεδιασμός της διπλωματικής εργασίας	6
2 Μεθοδολογία	7
2.1 Υπόβαθρο-συμβολισμοί	7
2.2 Αλγόριθμοι Κατάταξης και Ανίχνευσης Κοινοτήτων	8
2.2.1 PageRank	8
2.2.2 Label Propagation	9
2.3 CutBased Αλγόριθμοι	10
2.3.1 Υπολογισμός k-edge-out και k-vertex-out	11
2.3.2 Αλγόριθμος kCriticalNodes	14
2.3.3 Αλγόριθμος CutBased	16
2.3.4 Αλγόριθμος CutBase-Degree	18
2.3.5 Αλγόριθμοι CutBased1+ και CutBased2+	19
2.4 Οργάνωση πειραμάτων	24

3	Πειραματική Ανάλυση	26
3.1	Σύνολα δεδομένων	26
3.2	Πειραματική ανάλυση	27
3.2.1	Διαδικτυακά γραφήματα	27
3.2.2	Γραφήματα πραγματικών δεδομένων	31
3.2.3	Γραφήματα κοινωνικών δικτύων	35
4	Επίλογος	38
	Βιβλιογραφία	39
A	Απόδειξη χρόνου εκτέλεσης του αλγορίθμου CutBased	42
B	Αποτέλεσμα για όλα τα πειράματα	44
B.1	Πίνακες για NotreDame	44
B.2	Πίνακες για Food	48
B.3	Πίνακες για Google	51
B.4	Πίνακες για LastFm Asia	55
B.5	Πίνακες για Rome	58
B.6	Πίνακες για Twitter	62
B.7	Πίνακες για Gnutella25	65

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

2.1	Ροή εργασίας πειραμάτων	24
3.1	NorthDame αποτέλεσμα	28
3.2	Google αποτέλεσμα	30
3.3	Gnutella25 αποτέλεσμα	32
3.4	Rome αποτέλεσμα	34
3.5	Twitter αποτέλεσμα	36
3.6	LastFm Asia αποτέλεσμα	37

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

- 2.1 Μια επισκόπηση των αλγορίθμων που εξετάστηκαν. Τα όρια αναφέρονται με n κορυφές και m ακμές. Το αναφερόμενα όρια για τους LabelPropagation και PageRank υποθέτουν ότι εκτελούνται για σταθερό αριθμό επαναλήψεων 23
- 3.1 Τα γραφήματα που χρησιμοποιήθηκαν στα πειράματα, τα n και m είναι οι αριθμός των κορυφών και των ακμών, αντίστοιχα, SCC_s είναι ο αριθμός των ισχυρά συνδεδεμένων συνιστωσών, και $f(G)$ είναι το μέγεθος του μεταβατικού κλεισίματος. 27
- B.1 Αποτελέσματα της εκτέλεσης των αλγορίθμων για την αφαίρεση διαφορετικών ποσοστών κορυφών του NotreDame. Το $f(G)$ αναφέρεται στις υπόλοιπες τιμές της αρχικής τιμής transitive closure του G . $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές, και Time είναι ο χρόνος εκτέλεσης σε δευτερόλεπτα. . . . 44
- B.2 Αποτελέσματα της εκτέλεσης των αλγορίθμων για την αφαίρεση διαφορετικών ποσοστών κορυφών του Food. Το $f(G)$ αναφέρεται στις υπόλοιπες τιμές της αρχικής τιμής transitive closure του G . $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές, και Time είναι ο χρόνος εκτέλεσης σε δευτερόλεπτα. . . . 48
- B.3 Αποτελέσματα της εκτέλεσης των αλγορίθμων για την αφαίρεση διαφορετικών ποσοστών κορυφών του Google. Το $f(G)$ αναφέρεται στις υπόλοιπες τιμές της αρχικής τιμής transitive closure του G . $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές, και Time είναι ο χρόνος εκτέλεσης σε δευτερόλεπτα. . . . 51

- B.4 Αποτελέσματα της εκτέλεσης των αλγορίθμων για την αφαίρεση διαφορετικών ποσοστών κορυφών του LastFm Asia. Το $f(G)$ αναφέρεται στις υπόλοιπες τιμές της αρχικής τιμής transitive closure του G . $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές, και Time είναι ο χρόνος εκτέλεσης σε δευτερόλεπτα. . . . 55
- B.5 Αποτελέσματα της εκτέλεσης των αλγορίθμων για την αφαίρεση διαφορετικών ποσοστών κορυφών του Rome. Το $f(G)$ αναφέρεται στις υπόλοιπες τιμές της αρχικής τιμής transitive closure του G . $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές, και Time είναι ο χρόνος εκτέλεσης σε δευτερόλεπτα. . . . 58
- B.6 Αποτελέσματα της εκτέλεσης των αλγορίθμων για την αφαίρεση διαφορετικών ποσοστών κορυφών του Twitter. Το $f(G)$ αναφέρεται στις υπόλοιπες τιμές της αρχικής τιμής transitive closure του G . $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές, και Time είναι ο χρόνος εκτέλεσης σε δευτερόλεπτα. . . . 62
- B.7 Αποτελέσματα της εκτέλεσης των αλγορίθμων για την αφαίρεση διαφορετικών ποσοστών κορυφών του Gnutella25. Το $f(G)$ αναφέρεται στις υπόλοιπες τιμές της αρχικής τιμής transitive closure του G . $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές, και Time είναι ο χρόνος εκτέλεσης σε δευτερόλεπτα. . . . 65

ΚΑΤΑΛΟΓΟΣ ΑΛΓΟΡΙΘΜΩΝ

2.1	PageRank	9
2.2	Label Propagation	10
2.3	$k\text{EdgeOut}(G, s, k, R)$	13
2.4	Cut-exploration	15
2.5	Cut-selection	16
2.6	CutBased	18
2.7	$k\text{EdgeOutWithLocal1+}(G, s, k, R)$	20
2.8	$k\text{EdgeOutWithLocal2+}(G, s, k, R)$	21

ΠΕΡΙΛΗΨΗ

Φώτιος Γραμμένος, Δ.Μ.Σ. στη Μηχανική Δεδομένων και Υπολογιστικών Συστημάτων, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Ιωαννίνων, 2024.

Υπολογισμός κρίσιμων κόμβων σε κατευθυνόμενα γραφήματα.

Επιβλέπων: Λουκάς Γεωργιάδης, Αναπληρωτής Καθηγητής..

Στην παρούσα εργασία, διεξάγουμε μια εμπειριστατωμένη συγκριτική ανάλυση δύο θεμελιωδών προβλημάτων σε γραφήματα: του εντοπισμού κρίσιμων κόμβων σε ένα δίκτυο και του υπολογισμού της συνδεσιμότητας κάθε κορυφής. Συγκεκριμένα, εστιάζουμε στο πρόβλημα της εύρεσης κρίσιμων κόμβων, των οποίων η αφαίρεση οδηγεί στον μέγιστο δυνατό κατακερματισμό του αρχικού κατευθυνόμενου γραφήματος. Στόχος μας, είναι να υπολογίσουμε ένα ελάχιστο σύνολο κρίσιμων κόμβων που, όταν αφαιρούνται, ελαχιστοποιούν μια προκαθορισμένη συνάρτηση συνδεσιμότητας. Πιο συγκεκριμένα, δεδομένου ενός γραφήματος G και ενός ακέραιου αριθμού nc , ο στόχος είναι να βρεθεί ένα υποσύνολο S του συνόλου κορυφών V , που περιέχει το πολύ nc κόμβους, έτσι ώστε η συνάρτηση συνδεσιμότητας $f(G \setminus S)$ να ελαχιστοποιείται, αντιπροσωπεύοντας έτσι ένα συγκεκριμένο μέτρο κατακερματισμού. Ειδικότερα, όταν $nc = 1$, ο στόχος είναι να εντοπιστεί μια κορυφή x στην V τέτοια ώστε η $f(G \setminus x)$ να ελαχιστοποιείται, χαρακτηρίζοντας την εν λόγω κορυφή x ως τον πιο κρίσιμο κόμβο στο G . Παρέχουμε υλοποιήσεις των αξιολογημένων αλγορίθμων και παρουσιάζουμε μια εκτεταμένη πειραματική μελέτη τόσο σε πραγματικές όσο και σε τεχνητά δεδομένα. Κατά συνέπεια, η παρούσα εργασία αποσκοπεί στη σύγκριση ταχύτερων αλγορίθμων προσέγγισης και πρακτικών ευρετικών μεθόδων που λειτουργούν σε σχεδόν γραμμικό χρόνο. Εφαρμόζοντας το άπληστο αλγοριθμικό πλαίσιο μέχρι να αφαιρεθούν nc κορυφές, αναπτύσσουμε μια αποτελεσματική ευρετική μέθοδο για τη γενική περίπτωση, η οποία λειτουργεί σε $O(nc(m+n))$ χρόνο [1].

EXTENDED ABSTRACT

Fotios Grammenos, M.Sc. in Data and Computer Systems Engineering, Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Greece, 2024.

Computing critical nodes in Directed graphs.

Advisor: Loukas Georgiadis, Associate Professor..

The computation of critical nodes in directed graphs is a significant area of research with diverse applications across computer science, telecommunications, biology, computer security, network analysis, and other fields. A directed graph consists of nodes and directed edges that represent relationships or flows between nodes. A critical node is one whose removal significantly alters the structure or function of the graph, impacting connectivity or the efficiency of information flow.

In this MSc thesis, we conduct an in-depth comparative analysis of two fundamental problems: identifying critical nodes in a network and calculating the connectivity of each vertex. Specifically, we address the problem of identifying critical nodes whose removal results in the maximum possible fragmentation of the original directed graph. Our goal is to compute a minimal set of critical nodes that, when removed, minimize a predefined connectivity function. More precisely, given a graph G and an integer nc , we aim to find a subset S of the vertex set V , containing at most nc nodes, such that the connectivity function $f(G \setminus S)$ is minimized, thereby indicating a certain level of fragmentation. When $nc = 1$, the objective is to identify a vertex x in V such that $f(G \setminus x)$ is minimized, designating this vertex x as the most critical node in G .

We provide implementations of the evaluated algorithms and present an extensive experimental study on both real and artificial graphs. Consequently, this paper aims to compare faster approximation algorithms and practical heuristics that operate in near-linear time. By applying the greedy algorithmic framework until nc vertices are

removed, we develop an efficient heuristic method for the general case, which runs in $O(nc(m+n))$ time.

In particular, we have conducted a comparative analysis of several methods for computing critical nodes, including PageRank, Label Propagation, CutBased, CutBased-Degree, and two additional variations of CutBased. We also combine the algorithms Local1+ and Local2+ from the article [2] with those of CutBased. Our tests show that CutBased-Degree performs the best according to the connectivity function but has significantly higher computational time. Generally, we observe that in many of our experiments, CutBased1+ and CutBased2+ exhibit similar or slightly lower efficiency compared to CutBased and CutBased-Degree but achieve significantly faster processing times. On the other hand, PageRank and Label Propagation are more effective at maintaining the network structure compared to the cut-based methods.

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

- 1.1 Κίνητρο - Στόχοι
 - 1.2 Σχετική δουλειά
 - 1.3 Συνεισφορά
 - 1.4 Σχεδιασμός της διπλωματικής εργασίας
-

1.1 Κίνητρο - Στόχοι

Είναι γνωστό ότι, ο υπολογισμός των κρίσιμων κόμβων σε κατευθυνόμενα γραφήματα αποτελεί ένα σημαντικό πεδίο έρευνας με ποικίλες εφαρμογές στους τομείς της πληροφορικής, των τηλεπικοινωνιών, της βιολογίας και πολλών άλλων όπως στην ασφάλεια υπολογιστικών συστημάτων ή στην ανάλυση δικτύων. Ένα κατευθυνόμενο γράφημα (directed graph) αποτελείται από κόμβους (nodes) και κατευθυνόμενες ακμές (edges) που αντιπροσωπεύουν σχέσεις ή ροές μεταξύ των κόμβων. Ως κρίσιμος κόμβος (critical nodes) θεωρείται αυτός όπου η αφαίρεσή του προκαλεί σημαντική αλλαγή στη δομή ή στη λειτουργία του γραφήματος, επηρεάζοντας τη συνδεσιμότητα ή την αποτελεσματικότητα της ροής της πληροφορίας.

Επιπλέον, η αναγνώριση των κρίσιμων κόμβων είναι θεμελιώδης για την κατανόηση και τη βελτίωση της ανθεκτικότητας και της απόδοσης των συστημάτων που αναπαρίστανται με γραφήματα. Για παράδειγμα, στον τομέα της ασφάλειας δικτύων, η ταυτοποίηση και η προστασία των κρίσιμων κόμβων μπορεί να αποτρέψει

καταστροφικές επιθέσεις ή να ελαχιστοποιήσει τον αντίκτυπο τους. Στα κοινωνικά δίκτυα [3], οι κρίσιμοι κόμβοι μπορεί να αντιπροσωπεύουν άτομα με μεγάλη επιρροή και η ανάλυση τους μπορεί να βοηθήσει στην κατανόηση της διάδοσης της πληροφορίας ή της επιρροής σε ένα δίκτυο. Από την άλλη, στη βιολογία, οι κρίσιμοι κόμβοι σε μοριακά δίκτυα μπορούν να αντιπροσωπεύουν σημαντικά γονίδια ή πρωτεΐνες που είναι καθοριστικές για τις βιολογικές λειτουργίες [4].

Οι μέθοδοι υπολογισμού των κρίσιμων κόμβων περιλαμβάνουν την ανάλυση κεντρικότητας (centrality analysis) [5], την ανάλυση αρθρωτών σημείων (articulation points) και γεφυρών (bridges) [6], καθώς και άλλες τεχνικές που εξετάζουν την επίδραση της αφαίρεσης κόμβων στη συνολική συνδεσιμότητα και απόδοση του γραφήματος. Η χρήση αυτών των μεθόδων επιτρέπει την κατηγοριοποίηση και την αξιολόγηση των κόμβων με βάση τη σημασία τους, παρέχοντας πολύτιμες πληροφορίες για τη διαχείριση και τη βελτιστοποίηση σύνθετων συστημάτων.

Στο πλαίσιο λοιπόν, αυτής της εργασίας, εμβαθύνουμε στη συγκριτική ανάλυση δύο βασικών τεχνικών: την ανίχνευση κρίσιμων κόμβων ενός δικτύου και τον υπολογισμό της συνδεσιμότητας κάθε κορυφής. Πιο συγκεκριμένα, μελετάμε το πρόβλημα της ανίχνευσης κρίσιμων κόμβων με τέτοιο τρόπο ώστε η διαγραφή τους να προκαλεί τον μεγαλύτερο δυνατό κατακερματισμό του αρχικού κατευθυνόμενου γραφήματος. Ο στόχος της παρούσας εργασίας, είναι να υπολογιστεί ένα ελάχιστο σύνολο κρίσιμων κόμβων που, όταν αφαιρεθούν, ελαχιστοποιούν μια προκαθορισμένη συνάρτηση συνδεσιμότητας. Αναλυτικότερα, χρησιμοποιώντας πιο ακριβείς όρους, δεδομένου ενός γραφήματος G και ενός ακέραιου αριθμού nc , ο στόχος είναι να προσδιοριστεί ένα υποσύνολο S από το σύνολο κορυφών V , το οποίο θα περιέχει το πολύ nc κόμβους, έτσι ώστε η συνάρτηση συνδεσιμότητας $f(G \setminus S)$ να ελαχιστοποιείται, αντιπροσωπεύοντας ένα ορισμένο μέτρο κατακερματισμού. Συγκεκριμένα, όταν $nc = 1$, ο στόχος είναι να εντοπιστεί μια κορυφή x στο V όπου η $f(G \setminus x)$ είναι ελάχιστη, χαρακτηρίζοντας αυτή την κορυφή x ως τον πιο κρίσιμο κόμβο του G .

Επιπλέον, διάφορες πειραματικές έρευνες που έχουν υλοποιηθεί, επιβεβαιώνουν ότι αφαιρώντας κρίσιμους κόμβους, με την χρήση μερικών δημοφιλών μεθόδων όπως ο Page Rank, ο Max Degree, και ο Label Propagation, μειώνεται η συνάρτηση $f(G)$. Παρατηρήθηκε, λοιπόν ότι, η μέθοδος Label Propagation είναι η πιο αποτελεσματική ευρετική μέθοδος για διαδικτυακά γραφήματα (web graph).

Επιπρόσθετα, όπως σημειώνεται στα [7, 8], η ανίχνευση κρίσιμων κόμβων έχει αρκετές εφαρμογές στην ανάλυση κοινωνικών δικτύων. Αυτό έχει αποτελέσει αντι-

κείμενο σημαντικής έρευνας, με στόχο την καλύτερη κατανόηση διάφορων ιδιοτήτων όπως η συνοχή, η μεταβατικότητα, η κεντρική θέση των κόμβων και πολλά άλλα.

Από την άλλη πλευρά, ο υπολογισμός της συνδεσιμότητας κορυφής σε γραφήματα στοχεύει στον προσδιορισμό του ελάχιστου συνόλου κορυφών, των οποίων η αφαίρεση αποσυνδέει το γράφημα ή το αφήνει ως μονοσυνδεδεμένο. Αυτή η διαδικασία ιστορικά έχει πολυπλοκότητα $O(n^2)$ όταν $k = O(1)$, όπου k είναι η συνδεσιμότητα κορυφής του γραφήματος. Πρόσφατα κάποιοι πιθανοτικοί αλγόριθμοι έχουν επιτύχει σχεδόν γραμμική χρονική πολυπλοκότητα, σημειώνοντας σημαντική πρόοδο προς την κατεύθυνση μια λύσης γραμμικού χρόνου.[8]

Σύμφωνα με πειράματα του των Paudel et al. [4], αποδείχθηκε ότι η στρατηγική του, οδήγησε σε έναν αλγόριθμο γραμμικού χρόνου για τον υπολογισμό του πιο κρίσιμου κόμβου του G όσον αφορά την ισχυρή συνδεσιμότητα ανά ζεύγη. Οι ισχυρά συνδεδεμένες συνιστώσες ενός κατευθυνόμενου γραφήματος G διαχωρίζουν τις κορυφές του G με τέτοιο τρόπο ώστε δύο κορυφές x και y (ανήκουν στο σύνολο κορυφών V) να είναι ισχυρά συνδεδεμένες (συμβολίζεται ως $x \leftrightarrow y$) αν ανήκουν στην ίδια ισχυρά συνδεδεμένη συνιστώσα του G , δηλαδή αν υπάρχει κατευθυνόμενο μονοπάτι από τη μία στην άλλη και αντίστροφα. Ο υπολογισμός των ισχυρά συνδεδεμένων συνιστωσών ενός κατευθυνόμενου γραφήματος αποτελεί ένα από τα πιο βασικά προβλήματα των γραφημάτων, με πληθώρα εφαρμογών σε πολλούς διαφορετικούς τομείς. Επίσης, αυτή η στρατηγική που προαναφέρθηκε, προσφέρει μεγαλύτερη μείωση του αριθμού των ισχυρά συνδεδεμένων ζευγών κορυφών σε σύγκριση με άλλες γνωστές ευρετικές μεθόδους. Ο συγκεκριμένος αλγόριθμος υπολογίζει επαναληπτικά τον πιο κρίσιμο κόμβο και στη συνέχεια τον αφαιρεί.

Ως εκ τούτου, στη συγκεκριμένη εργασία, στοχεύουμε να συγκρίνουμε ταχύτερους αλγόριθμους προσέγγισης και πρακτικές ευρετικές μεθόδους που λειτουργούν σε σχεδόν γραμμικό χρόνο. Εφαρμόζουμε το αλγοριθμικό πλαίσιο απληστίας (the greedy algorithmic framework), μέχρι να αφαιρέσουμε nc κορυφές και να επιτύχουμε μια αποτελεσματική ευρετική μέθοδο για τη γενική περίπτωση, η οποία εκτελείται σε χρόνο $O(nc(m + n))$.

1.2 Σχετική δουλειά

Το πρόβλημα της ανίχνευσης κρίσιμων κόμβων έχει μελετηθεί για διάφορες συναρτήσεις συνδεσιμότητας, κυρίως όμως σε μη κατευθυνόμενα γραφήματα. Όπως αναλύεται στην αναφορά [8], ο εντοπισμός κρίσιμων κόμβων αποτελεί έναν αποδοτικό τρόπο για την ανάλυση και κατανόηση των ιδιοτήτων, των δομών και των λειτουργιών των δικτύων. Αυτό διευκολύνει τον έλεγχο του δικτύου, είτε ο στόχος είναι η συντήρηση του, είτε η αποδόμηση του, καθώς οι κρίσιμοι κόμβοι είναι αυτοί που διατηρούν τη συνοχή του, ενώ η αφαίρεση τους μειώνει σημαντικά τη συνδεσιμότητα του. Το συγκεκριμένο άρθρο εστιάζει κυρίως στη συνδυαστική πτυχή του προβλήματος. Κυρίως, εξετάζει και αναφέρει διάφορες πρόσφατες εξελίξεις και ευρήματα που έχουν προκύψει στη βιβλιογραφία για κάθε παραλλαγή. Αυτό περιλαμβάνει τη θεωρητική πολυπλοκότητα, τους ακριβείς αλγορίθμους, τα συστήματα προσέγγισης, τους ευρετικούς αλγορίθμους και την εξαγωγή νέων αποτελεσμάτων. Αναφέρεται επίσης ότι μερικά απλά προβλήματα αναζήτησης επιλύονται με την χρήση αλγορίθμων πολυωνυμικού χρόνου. Αυτά τα προβλήματα ανήκουν στην κατηγορία P , ενώ τα προβλήματα με αλγορίθμους επαλήθευσης πολυωνυμικού χρόνου αποτελούν την κλάση NP . Αντίθετα, για την επίλυση δύσκολων προβλημάτων, γίνεται χρήση άπληστων αλγορίθμων.

Έπειτα, οι συγγραφείς του άρθρου [9] μελέτησαν πειραματικά το πώς μειώνεται το $f(G)$ μετά την αφαίρεση κρίσιμων κόμβων σε κοινωνικά και διαδικτυακά γραφήματα, όταν αυτοί οι κρίσιμοι κόμβοι επιλέγονται με τη χρήση ορισμένων δημοφιλών ευριστικών, όπως η Betweenness Centrality [10], Page Rank [11], Max Degree και Label Propagation [12]. Πιο συγκεκριμένα, εξετάζεται μια φυσική συνάρτηση συνδεσιμότητας (connectivity function), η οποία μετρά τον αριθμό των (διατεταγμένων) ζευγών κορυφών (u, v) , ώστε το v να είναι προσβάσιμο από το u , δηλαδή, $f(G) = \sum_{u \in V} R(u)$, με $R(u)$ το σύνολο των κορυφών που είναι προσβάσιμες από το u (συμπεριλαμβανομένου του u).

Σχετικά με την αναφορά [11], αναλύεται ο αλγόριθμος Page Rank, μια μέθοδος για την αντικειμενική και αυτοματοποιημένη αξιολόγηση ιστοσελίδων. Συγκρίνει το PageRank με έναν ιδανικό τυχαίο περιηγητή στον Παγκόσμιο Ιστό, δείχνοντας πώς να υπολογίζουμε εύκολα το PageRank για μεγάλο αριθμό σελίδων και πώς να το εφαρμόζουμε στην αναζήτηση και πλοήγηση χρηστών. Προκειμένου να υλοποιηθεί αυτό, προτείνεται το PageRank, μια μέθοδος υπολογισμού μιας κατάταξης για

κάθε ιστοσελίδα με βάση το γράφημα του ιστού. Το PageRank έχει εφαρμογές στην αναζήτηση, την περιήγηση και την εκτίμηση της διαδρομής.

Συνεχίζοντας, με τον αλγόριθμο Label Propagation, στο [12] εξετάστηκε ο συγκεκριμένος αλγόριθμος με χρήση μόνο της δομής του δικτύου και καμία αντικειμενική συνάρτηση βελτιστοποίησης, ούτε προκαταρκτικές πληροφορίες για τις κοινότητες. Αποδεικνύεται ότι, ο αλγόριθμος απαιτεί σχεδόν γραμμικό χρόνο, καθιστώντας τον υπολογιστικά λιγότερο δαπανηρό από ό,τι ήταν δυνατό μέχρι σήμερα. Μπορούμε να θεωρήσουμε ότι ο Label Propagation είναι η πιο αποτελεσματική ευρετική μέθοδος για τους διαδικτυακά γραφήματα.

Άλλος ένας αλγόριθμος, ο οποίος χρησιμοποιείται στη θεωρία γραφημάτων και στην ανάλυση δικτύων για τον εντοπισμό σημαντικών ακμών σε ένα γράφημα, είναι ο CutBased και ο CutBased-Degree αντίστοιχα. Είναι ιδιαίτερα χρήσιμοι σε εφαρμογές όπως η ανίχνευση κοινοτήτων, η αποσύνθεση δικτύων και η κατανόηση των δομικών ιδιοτήτων πολύπλοκων δικτύων. Στην αναφορά [13] λοιπόν, μελετιούνται δύο εκδοχές του προβλήματος: το πρόβλημα χωρίς περιορισμούς και το θεμελιώδες πρόβλημα cut bases. Στην εκδοχή χωρίς περιορισμούς, όπου οι περικοπές μπορούν να είναι οποιουδήποτε είδους, το πρόβλημα μπορεί να διατυπωθεί ως πρόβλημα ροής πολλαπλών τερματικών σε δίκτυο, και έτσι μπορεί να επιλυθεί σε πολυωνυμικό χρόνο. Αντίθετα, στο θεμελιώδες πρόβλημα της cut base, όπου όλες οι κοπές προκύπτουν με τη διαγραφή μιας ακμής από ένα δέντρο T , αποδεικνύεται ότι είναι NP -δυσχερές. Επίσης, αποδεικνύεται ότι ένα δέντρο που προκαλεί την ελάχιστη θεμελιώδη βάση κύκλων είναι και μια βέλτιστη λύση για το πρόβλημα της ελάχιστης θεμελιώδους cut base σε μη σταθμισμένα γραφήματα. Επιπρόσθετα, συγκρίνοντας τους δύο αλγόριθμους, ο CutBased και ο CutBased-Degree, παρατηρούμε ότι διαφέρουν στον τρόπο που καθορίζουν τη κρισιμότητα των κόμβων και ακμών σε ένα δίκτυο. Αρχικά, ο Cut-Based Algorithm εστιάζει κυρίως στις ακμές και την αποσύνθεση του δικτύου μέσω αυτών, ενώ ο CutBased-Degree συνδυάζει την ανάλυση των ακμών με την ανάλυση του βαθμού των κόμβων. Από την άλλη, ο CutBased-Degree μπορεί να παρέχει πιο λεπτομερείς πληροφορίες για τη σημαντικότητα τόσο των ακμών όσο και των κόμβων, ενώ ο Cut-Based Algorithm είναι πιο κατάλληλος για ανάλυση κοινοτήτων και ανθεκτικότητας δικτύων.

Τέλος, αξίζει να σημειωθεί ο αλγόριθμος Local EC (Local1+, Local2+), ο οποίος σύμφωνα με το άρθρο [2] είναι ένας πολύ αποτελεσματικός υπολογισμός της συνδεσιμότητας των κορυφών. Σε αυτό το άρθρο μελετάται η απόδοση των αλγορίθμων

σχεδόν γραμμικού χρόνου της αναφοράς [14] για μικρή συνδεσιμότητα κορυφών. Επισημαίνεται μια πιθανή βελτίωση του πλαισίου, ενώ παράλληλα προσθέτουν μια νέα ευρετική μέθοδο, όπου ονομάζεται ανίχνευση τοπικών αποκοπών (local cut detection), την οποία αποκαλούν ως μέτρηση βαθμού (degree counting).

1.3 Συνεισφορά

Η παρούσα εργασία συνεισφέρει στην κατανόηση και την επίλυση του προβλήματος εντοπισμού κρίσιμων κόμβων σε κατευθυνόμενα γραφήματα. Συγκεκριμένα, έχει διεξάγει μια συγκριτική ανάλυση διάφορων μεθόδων υπολογισμού κρίσιμων κόμβων, συμπεριλαμβανομένων των μεθόδων Page Rank, Label Propagation, Cut-Based, CutBased-Degree. Αναλυτικότερα, πραγματοποιήθηκαν πειραματικές δοκιμές χρησιμοποιώντας πραγματικά δεδομένα από κοινωνικά δίκτυα και διαδικτυακά γραφήματα, αναδεικνύοντας την αποτελεσματικότητα και την αποδοτικότητα κάθε μεθόδου, αλλά και τα πλεονεκτήματα και τα μειονεκτήματα αυτών. Επίσης συνδιάζει τους αλγόριθμους που Local1+ και Local2+ από το άρθρο [2], με αυτούς του CutBased. Μέσα από την σύγκριση των προτεινόμενων μεθόδων και των αποτελεσμάτων, αναπαράγονται πολύτιμες γνώσεις για την ανάλυση και τη βελτιστοποίηση των αναφερόμενων αλγορίθμων.

1.4 Σχεδιασμός της διπλωματικής εργασίας

Το Κεφάλαιο 2 παρέχει τις θεμελιώδεις πληροφορίες σχετικά με το θεωρητικό υπόβαθρο, τους συμβολισμούς και τους αλγορίθμους. Στο Κεφάλαιο 3 παρουσιάζονται τα αποτελέσματα των πειραμάτων μας, ενώ το Κεφάλαιο 4 περιλαμβάνει τον επίλογο της εργασίας. Στα Παραρτήματα, μπορείτε να βρείτε λεπτομερείς αποδείξεις για το χρόνο εκτέλεσης των αλγορίθμων που βασίζονται στον Cutbased, καθώς και αναλυτικούς πίνακες με τα αποτελέσματα των πειραμάτων.

ΚΕΦΑΛΑΙΟ 2

ΜΕΘΟΔΟΛΟΓΙΑ

2.1 Υπόβαθρο-συμβολισμοί

2.2 Αλγόριθμοι Κατάταξης και Ανίχνευσης Κοινοτήτων

2.3 CutBased Αλγόριθμοι

2.4 Οργάνωση πειραμάτων

Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, στόχος της παρούσας εργασίας είναι η σύγκριση και η ανάλυση πειραματικών αποτελεσμάτων, όπου προέκυψαν από αλγόριθμους, οι οποίοι αναζητούν-εντοπίζουν και αφαιρούν ένα σύνολο κρίσιμων κόμβων. Αναλυτικότερα, σε αυτό το κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο, όπου περιέχει απαραίτητους συμβολισμούς και ορισμούς, αναπτύσσονται οι αλγόριθμοι που χρησιμοποιήθηκαν και τέλος περιγράφεται τόσο το περιβάλλον υλοποίησης, όσο και η μεθοδολογία που ακολουθήθηκε για την διεξαγωγή των συγκεκριμένων πειραμάτων.

2.1 Υπόβαθρο-συμβολισμοί

Έστω $G = (V, E)$ ένας κατευθυνόμενος γράφημα (digraph). Για κάθε υπογράφημα H του γράφημα G , συμβολίζουμε με $V(H) \subseteq V$, το σύνολο των κορυφών του H , και με $E(H) \subseteq E$, το σύνολο των ακμών του H . Για $S \subseteq V$, συμβολίζουμε με $G(S)$ το υπογράφημα του G που αποτελείται από τις κορυφές του S . Ενώ το υπογράφημα

$G - S$ προκύπτει έπειτα από την αφαίρεση των κορυφών του συνόλου S από το γράφημα G .

Έχοντας και πάλι ένα κατευθυνόμενο γράφημα $G = (V, E)$ όπου $s \in V$ είναι μια κορυφή. Η k -edge-out συνιστώσα του s είναι ένα ελάχιστο υπογράφημα S του G , που περιλαμβάνει το s και έχει το πολύ k εξερχόμενες ακμές προς κορυφές εκτός του S στο $G \setminus S$. Από την άλλη, η k -vertex-out συνιστώσα του s είναι ένα ελάχιστο υπογράφημα S του G , που περιλαμβάνει το s και έχει το πολύ k κορυφές $X \subseteq V(S)$, με εξερχόμενες ακμές προς κορυφές εκτός του S στο $G \setminus S$.

Δεδομένου ενός κατευθυνόμενου γραφήματος $G = (V, E)$, η μεταβατική κλειστότητα του G , είναι ένα γράφημα $G' = (V, E')$ τέτοιο ώστε για κάθε ζεύγος κορυφών $(u, v) \in E$ να υπάρχει έστω μια ακμή $(u, v) \in E'$, αν και μόνο αν, υπάρχει κατευθυνόμενη διαδρομή από το u προς v στο G . Με άλλα λόγια, η μεταβατική κλειστότητα του G προσθέτει ακμές για να διασφαλίσει ότι αν υπάρχει τρόπος να φτάσει κανείς στο u από το v μέσω μιας ακολουθίας ενδιάμεσων κορυφών, τότε υπάρχει μια απευθείας ακμή από το u στο v , εντός του G' . Η κατανόηση της μεταβατικής κλειστότητας βοηθά στην ανάλυση των ευρύτερων επιπτώσεων της συνδεσιμότητας και του ρόλου των κρίσιμων κόμβων στη διατήρηση μας. Επιπλέον, η μεταβατική κλειστότητα του γραφου G μετά την αφαίρεση του συνόλου S συμβολίζεται ως $f(G \setminus S)$.

2.2 Αλγόριθμοι Κατάταξης και Ανίχνευσης Κοινοτήτων

2.2.1 PageRank

PageRank (PR) είναι ένας γνωστός αλγόριθμος που χρησιμοποιείται για την κατάταξη των ιστότοπων, όπως προαναφέρθηκε. Ωστόσο, είναι ικανός να βρίσκει κρίσιμες κορυφές σε έναν γράφο. Προέρχεται από το έργο των Page et al. [11] και η βασική ιδέα είναι η εξής: μια σελίδα (κορυφή x) έχει υψηλή βαθμολογία κατάταξης σελίδας εάν το άθροισμα των βαθμολογιών των σελίδων (κορυφών) που συνδέονται (με out-edge) με την τρέχουσα σελίδα είναι επίσης υψηλό. Ο αλγόριθμος ξεκινά με την ανάθεση ενός διανύσματος με ίση κατανομή τιμών, ως αρχική βαθμολογία. Στη συνέχεια, μέχρι να επιτύχει σύγκλιση ή να φτάσει σε έναν προκαθορισμένο μέγιστο αριθμό επαναλήψεων, το διάνυσμα ενημερώνεται με βάση την ακόλουθη εξίσωση:

$$pr(j+1) = (1-b) \cdot \frac{1}{n} + b \cdot pr(j) \cdot M;$$

όπου b είναι η σταθερά τηλεμεταφοράς (όπου στην δικιά μας περίπτωση ισούται με 0,85), με M ο τυχαίος πίνακας μετάβασης, δηλαδή $M = D - A$, όπου D ο διαγώνιος πίνακας των βαθμών εξόδου και A ο πίνακας γειτνίασης. Ο χρόνος εκτέλεσης ανά επανάληψη είναι $O(m)$, όπου m είναι ο αριθμός των ακμών. Ένας απλός ψευδοκώδικας αυτού του αλγορίθμου παρατίθεται στον αλγόριθμο 2.1.

Αλγόριθμος 2.1 PageRank

Require: $G, n_{\text{percentages}}$, max number of iterations K , threshold e

- 1: $i \leftarrow 1$;
 - 2: **while** $i < n_{\text{percentages}}$ **do**
 - 3: Initialize $pr = [1/n, 1/n, \dots, 1/n]$ /* PageRank vector initialized to equal values */
 - 4: $j \leftarrow 1$;
 - 5: **for** $j < K$ and $|pr(j+i) - pr(j)| > e$ **do**
 - 6: $pr(j+1) = (1-b) \cdot 1/n + b \cdot pr(j) \cdot M$;
 - 7: $j \leftarrow j+1$;
 - 8: **end for**
 - 9: Select and delete the critical node as the one the maximum pr value.
 - 10: $i \leftarrow i+1$;
 - 11: **end while**
 - 12: Calculate the final transitive closure $T(G - S)$, where $S \subseteq V$ is the set of deleted critical nodes.
-

2.2.2 Label Propagation

Label Propagation (LP) είναι ένας αλγόριθμος που χρησιμοποιείται στο πρόβλημα της ανίχνευσης κοινότητας και προέρχεται από την εργασία των Raghavan et al. [12]. Η βασική ιδέα του LP είναι να «εξαπλώνονται» ετικέτες στο δίκτυο έως ότου να επιτευχθεί μια ισορροπία. Στην αρχή αυτού του αλγορίθμου, κάθε κορυφή συνδέεται με μια μοναδική ετικέτα. Κατά τη διάρκεια μιας επανάληψης, κάθε κορυφή του γραφήματος επεξεργάζεται με τυχαία σειρά και της αποδίδεται μια νέα ετικέτα, σύμφωνα με τις ετικέτες των γειτόνων της. Πιο συγκεκριμένα, λαμβάνει την ετικέτα

με τις περισσότερες εμφανίσεις μεταξύ των γειτόνων της. Η προηγούμενη διαδικασία επαναλαμβάνεται έως ότου δεν υπάρχουν αλλαγές στις ετικέτες του γραφήματος. Ο χρόνος εκτέλεσης ανά επανάληψη είναι $O(m)$, όπου m είναι ο αριθμός των ακμών. Στην περίπτωσή μας, μετά τον υπολογισμό των ετικετών των κορυφών, χωρίζουμε τις κορυφές σε κοινότητες και επιλέγουμε ως κρίσιμη κορυφή, αυτήν που έχει τον μέγιστο αριθμό γειτόνων με άλλες κοινότητες. Ένας απλός ψευδοκώδικας αυτού του αλγορίθμου παρατίθεται στον αλγόριθμο 2.2.

Αλγόριθμος 2.2 Label Propagation

Require: $G, n_{\text{percentages}}$

- 1: initialize the labels for all nodes in the graph.
 - 2: $i \leftarrow 1$;
 - 3: **while** $i < n_{\text{percentages}}$ **do**
 - 4: *randomize*($V(G)$);
 - 5: **for** $v \in V(G)$ **do**
 - 6: Find the most frequent label l among the neighbors of v .
 - 7: **if** the current label of v is not l **then**
 - 8: Update the label of v to l .
 - 9: **end if**
 - 10: **end for**
 - 11: Divide the nodes into communities based on their labels.
 - 12: Select and delete the critical node as the one with the maximum number of neighbors in other communities.
 - 13: $i \leftarrow i + 1$;
 - 14: **end while**
 - 15: Calculate the final transitive closure $T(G - S)$, where $S \subseteq V$ is the set of deleted critical nodes.
-

2.3 CutBased Αλγόριθμοι

Σε αυτή την υποενότητα θα αναλύσουμε μεθόδους όπου υπολογίζουν κρίσιμους κόμβους αλλά είναι βασισμένες στον αλγόριθμο CutBase. Γενικότερα, ένας αλγόριθμος τέτοιου είδους, υπολογίζει κρίσιμους κόμβους με σχεδόν γραμμικό χρόνο, όπου $k = O(1)$. Εφαρμόζει έναν άπληστο αλγόριθμο που καλεί επανειλημμένα τη

μέθοδο `kCriticalNodes` (2.3.2) μέχρι να διαγραφεί ο επιθυμητός αριθμός κρίσιμων κόμβων nc από το γραφήματα. Κατά τη διάρκεια του αλγορίθμου, χρησιμοποιούμε το $G(S)$ για να συμβολίσουμε το υπογράφημα του G , το οποίο αποτελείται από το σύνολο των κορυφών του S . Το $\text{vol}_{\text{out}}(S)$, ενός συνόλου κορυφών S ορίζεται ως ο αριθμός των ακμών που εξέρχονται από τις κορυφές στο S (δηλαδή, οι ακμές του G που έχουν ένα άκρο στο S και το άλλο στο $(V \setminus S)$). Επίσης, ένα σύνολο ακμών C είναι *edge-cut* (αντίστοιχα, ένα *vertex-cut*) εάν η αφαίρεση των ακμών (ή των κορυφών) του C από το G αφήνει μια κορυφή v απρόσιτη από άλλη κορυφή u , για κάθε ζεύγος κορυφών u, v όπου το v είναι προσβάσιμο από το u στο G . Ο `kCriticalNodes` γραμμικός αλγόριθμος χωρίζεται σε δύο στάδια:

- **Cut-exploration:** για κάθε δείγμα κορυφής s , εντοπίζει περικοπές περιορισμένου μεγέθους που εμφανίζονται σε όλες τις διαδρομές από το s προς άλλες κορυφές. Αυτό επιτυγχάνεται μέσω του αλγορίθμου `kEdgeOut`, όπου παραθέτεται ένας απλοποιημένος ψευδοκώδικας στον αλγόριθμο 2.3. Έπειτα, αντιστοιχίζεται στις κορυφές των περικοπών μια εκτίμηση για τον αριθμό των ζευγαριών που αποσυνδέουν. Ένας απλός ψευδοκώδικας αυτού του αλγορίθμου παρατίθεται στον αλγόριθμο 2.4.
- **Cut-selection:** Επαναλαμβάνονται όλες οι περικοπές που διερευνήθηκαν στο πρώτο στάδιο (αλγόριθμος 2.4) και επιλέγεται η τομή που αποσυνδέει τα περισσότερα μονοπάτια, σύμφωνα με τις εκτιμήσεις. Ένας απλός ψευδοκώδικας αυτού του αλγορίθμου παρατίθεται στον αλγόριθμο 2.5.

2.3.1 Υπολογισμός *k-edge-out* και *k-vertex-out*

Έχοντας στο μυαλό μας τον ορισμό του *k-edge-out* και του *k-vertex-out* που δόθηκε στην ενότητα 2.1, γνωρίζουμε ότι το δείγμα κορυφής s , μπορεί να έχει περισσότερες από μία συνιστώσες με *k-edge-out* (όπως αναφέρεται στο [15]). Επίσης, για $k' < k$, κάθε συνιστώσα με *k-edge-out* της κορυφής s είναι επίσης συνιστώσα με *k'-edge-out* της κορυφής s . Ο αλγόριθμος, αναζητεί τα υπογράφηματα S με *k-edge-out*, όπου οι ακμές αποσυνδέουν όλα τα ζεύγη κορυφών στο $S \times (V \setminus S)$. Οι εξής εγγυήσεις παρέχονται απο τον αλγόριθμο που υπολογίζει τέτοιες συνιστώσες:

- Εάν υπάρχει μια συνιστώσα με *k-edge-out* μιας κορυφής s με όγκο το πολύ Δ , τότε ο αλγόριθμος επιστρέφει ένα σύνολο με *k-edge-out* και όγκο $O(k\Delta)$

(το οποίο μπορεί να μην είναι ελάχιστο),

- Μπορεί να καταλήξει στο συμπέρασμα ότι δεν υπάρχει σύνολο με k -edge-out της κορυφής s με όγκο $O(\Delta)$.

Ο πρώτος αλγόριθμος για τον υπολογισμό μιας συνιστώσας με k -edge-out με τις παραπάνω εγγυήσεις παρουσιάστηκε στο [16] και εκτελείται σε χρόνο $O(\Delta k^{O(\Delta k)})$. Αργότερα, αυτό το όριο βελτιώθηκε χρησιμοποιώντας τυχαιοποίηση σε $O(\Delta k^2)$, όπως αποδεικνύεται στις αναφορές [14, 17, 18]. Στην συγκεκριμένη εργασία γίνεται χρήση του αλγόριθμου των Nanongkai et al. [17], ο οποίος συνεπάγεται μικρότερους σταθερούς παράγοντες στον χρόνο εκτέλεσης λόγω της απλότητάς του, για $k > 1$. Ο αλγόριθμος εντοπίζει σωστά μια συνιστώσα με k -edge-out και όγκο $O(\Delta k)$, ή συμπεραίνει ότι δεν υπάρχει συνιστώσα με k -edge-out και όγκο $O(\Delta)$ με πιθανότητα επιτυχίας $p \geq 1/2$. Δηλαδή, με πιθανότητα $1 - p$ ο αλγόριθμος συμπεραίνει εσφαλμένα ότι δεν υπάρχει συνιστώσα με k -edge-out και όγκο $O(\Delta)$. Εάν ο αλγόριθμος εντοπίσει μια συνιστώσα με k -edge-out, αυτή είναι εγγυημένα υπαρκτή, δηλαδή δεν υπάρχουν ψευδώς θετικά αποτελέσματα. Μπορεί κανείς να επαναλάβει την αναζήτηση $O(\log n)$ φορές για να εξασφαλίσει ότι η σωστή απάντηση υπολογίζεται με μεγάλη πιθανότητα, αλλά στην συγκεκριμένη περίπτωση επαναλαμβάνεται η αναζήτηση $O(1)$ φορές προκειμένου να παράγει ταχύτερους χρόνους εκτέλεσης. Για λόγους ευκολίας, συμβολίζουμε μια συνιστώσα S με k -edge-out ως $\langle S, C \rangle$, όπου C είναι το σύνολο των εξερχόμενων ακμών από το S , $\leq k$.

Να σημειωθεί ότι για να προσδιοριστεί μια συνιστώσα με k -edge-out μιας κορυφής s , πρέπει να οριστεί ένα ανώτερο όριο του όγκου Δ , όπως απαιτείται από την παραπάνω διαδικασία. Για να γίνει αυτό, αναζητάται μια συνιστώσα με k -edge-out και όγκο που αυξάνεται εκθετικά, ξεκινώντας από $\Delta = 2$ και μέχρι το $\Delta = R/k$, όπου R είναι ο αριθμός των προσπελάσιμων ακμών από το s . Έτσι, ο συνολικός χρόνος για τον εντοπισμό μιας συνιστώσας με k -edge-out και όγκο Δ εξακολουθεί να είναι $O(\Delta k^2)$. Στην περίπτωση που δεν υπάρχει συνιστώσα με k -edge-out της s , ή ο αλγόριθμος αποτυγχάνει να εντοπίσει μία, ο χρόνος εκτέλεσης είναι $O(Rk)$. Ο ψευδοκώδικας για τον υπολογισμό μιας συνιστώσας με k -edge-out εμφανίζεται στον Αλγόριθμο 2.3.

Ο υπολογισμός των k -vertex-out συνιστωσών μπορεί να αναχθεί στον υπολογισμό των k -edge-out συνιστωσών, σύμφωνα με την έρευνα [16]. Αρχικά πρέπει, να μετασχηματιστεί το γράφημα εισόδου $G = (V, E)$ σε ένα γράφημα $G' = (V', E')$,

Αλγόριθμος 2.3 $k\text{EdgeOut}(G, s, k, R)$

```
1: for  $i = \log(2k)$  to  $\log(R)$  do
2:    $(S, C) \leftarrow \emptyset$ .
3:    $\Delta \leftarrow \frac{2^i}{k}$ 
4:   for  $i = 1$  to  $k + 1$  do
5:     Execute a DFS from  $s$  visiting exactly  $k\Delta$  edges. Let  $T$  be the resulting DFS
       tree.
6:     if less than  $k\Delta$  edges were visited then
7:       Let  $S$  be the set of visited vertices by the last DFS traversal, and let  $C$  be
       the set of outgoing edges from  $S$ .
8:       Restore all reversed edges.
9:     end if
10:    Sample an edge  $(x, y)$  uniformly at random among the visited edges and
       reverse the path  $T[s, x]$ .
11:   end for
12:   if  $S \neq \emptyset$  then
13:     return  $(S, C)$ .
14:   end if
15:   Restore all reversed edges and  $(S, C) \leftarrow \emptyset$ .
16: end for
17: return  $(\emptyset, \emptyset)$ .
```

προσθέτοντας για κάθε κόμβο $v \in V$ δύο κόμβους v_{in} και v_{out} στο V' και μια ακμή $(v_{\text{in}}, v_{\text{out}})$ στο E , και για κάθε ακμή $(x, y) \in E$ προσθέτουμε μια ακμή $(x_{\text{out}}, y_{\text{in}})$ στο E . Παρατηρείται ότι ο αριθμός των κόμβων $n' = |V'| = 2|V| = O(n)$ και ο αριθμός των ακμών $m' = |E'| = |E| + |V| = O(m)$.

Επομένως, μπορεί κανείς να υπολογίσει τις k -vertex-out συνιστώσες στο G υπολογίζοντας τις k -edge-out συνιστώσες στο G' . Για ευκολία, εφαρμόζεται εξ ολοκλήρου στο γράφημα G' και αντιμετωπίζει τις διαγραφές κόμβων αναλόγως (δηλαδή όταν ο αλγόριθμος αποφασίζει να διαγράψει έναν κόμβο $v \in \{v_{\text{in}}, v_{\text{out}}\}$, διαγράφει και τους δύο v_{in} και v_{out}). Έστω v ένας κόμβος του V' που μπορεί να είναι είτε u_{in} είτε u_{out} για κάποιον κόμβο $u \in V$.

2.3.2 Αλγόριθμος kCriticalNodes

Έχοντας επεξηγήσει τα παραπάνω, μπορούμε να περιγράψουμε τον αλγόριθμο kCriticalNodes με μεγαλύτερη λεπτομέρεια. Στη φάση cut-exploration (2.4), ο αλγόριθμος παίρνει ως είσοδο ένα γράφημα G' , δειγματοληπτεί ένα τυχαίο κόμβο s , και επαναλαμβάνει την αναγνώριση ενός k -edge-out (S, C) . Έστω $r_n(s)$ και $r_m(s)$ ο αριθμός των κόμβων και των ακμών, αντίστοιχα, που είναι προσβάσιμοι από το s . Για κάθε αναγνωρισμένη k -edge-out (S, C) εκτιμάται ο αριθμός των ζευγών που γίνονται μη προσβάσιμα στο $G \setminus C$, και ενισχύονται οι κόμβοι στο $V(C)$, αυξάνοντας τις βαθμολογίες τους. Για αυτό το λόγο διατηρείται ένα βάρος $w[v]$ για κάθε κόμβο v (αρχικοποιημένο σε 1), το οποίο είναι, ο αριθμός των κόμβων που έχουν συμπεριστεί στο v και μπόρεσαν να φτάσουν από το $G[S]$ στο $V(C)$. Ορίζεται επίσης $w[S] = \sum_{v \in S} w[v]$ για $S \subseteq V$. Εάν βρεθεί ένας διαχωρισμός που διακόπτει όλα τα μονοπάτια από το v προς ένα σύνολο κόμβων S' , τότε ο ίδιος διαχωρισμός διακόπτει και όλα τα μονοπάτια από τους κόμβους $w[v]$ προς το σύνολο S' (τα οποία έχουν συμπεριστεί σε v). Για να διατηρηθούν τα βάρη, κάθε φορά που βρίσκεται μια (S, C) (k -edge-out συνιστώσα), όπου $s \in S$, υπολογίζεται το σύνολο $S' \subseteq S$ που φτάνει στο $V(C)$ στο $G[S]$, και μετά τη συρρίκνωση του S στο s ορίζεται $w[s] = w[S']$ (σημειώστε ότι $s \in S'$ καθώς το s μπορεί να φτάσει το $V(C)$). Αφού αναγνωριστεί μια (S, C) για το s , και μετά τη συμπίεση του s στο $S \cup V(C)$, εκτιμάται ότι ο αριθμός των ζευγών που δεν είναι πλέον προσβάσιμα μετά τη διαγραφή του C είναι $w[s] \cdot (r_n(s) - w[S'])$. Έτσι, αυξάνεται το σκορ $\text{score}[x]$ των αρχικών άκρων x των ακμών στο C κατά

$$\frac{w[s] \cdot (r_n(s) - w[S'])}{|C|}$$

καθώς αυτός είναι ο μέσος όρος των ζευγών που κάθε κορυφή της τομής διασπά. Ωστόσο, ακόμη κι αν οι εκτιμήσεις μας ήταν ακριβείς, δεν ισχύει ότι η αφαίρεση μιας κορυφής από το $V(C)$ θα αποσυνδέσει

$$\frac{w[s] \cdot (r_n(s) - w[S'])}{|C|}$$

ζεύγη, για $k > 1$, καθώς κάθε μεμονωμένη κορυφή δεν αποτελεί τομή για το s . Τέλος, ενημερώνονται οι αριθμοί $r_n(s)$ και $r_m(s)$ για να αντικατοπτρίσουν τον αριθμό των κορυφών και των ακμών που αφαιρέθηκαν από το γράφημα.

Στο δεύτερο βήμα του αλγορίθμου (Cut-selection 2.5), επιλέγεται η τομή με την υψηλότερη βαθμολογία και πραγματοποιείται η διαγραφή της ακμής με την υψηλότερη βαθμολογία για κάθε μία από τις ακμές της. Είναι σημαντικό να σημειωθεί

ότι αυτή η διαδικασία μπορεί να αφαιρέσει τομές διαφόρων μεγεθών (από 1 έως k), αλλά σε κάθε περίπτωση, διαγράφουμε το άκρο (κορυφή) που συνδέεται με την υψηλότερη βαθμολογία μεταξύ των άκρων κάθε ακμής τομής. Ένας απλός ψευδοκώδικας αυτού του αλγορίθμου παρατίθεται στον αλγόριθμο 2.4.

Αλγόριθμος 2.4 Cut-exploration

Require: G, n_{sample} (number of samples), k (number of nodes to select)

- 1: Initialize $score[v] = 0$ for all $v \in V$
 - 2: Set $C = \emptyset$
 - 3: **for** $i = 1$ **to** n_{sample} **do**
 - 4: Initialize $w[v] = 1$ for all $v \in V$
 - 5: Compute $r_n(s)$ and $r_m(s)$, the number of reachable vertices and edges from s
 - 6: **repeat**
 - 7: $(S, C) \leftarrow kEdgeOut(G, s, k, r_m(s))$
 - 8: **if** $C \neq \emptyset$ **then**
 - 9: Let Co be the original edges of C contractions
 - 10: Insert Co into C
 - 11: Compute S' , the set of vertices that can reach any vertex in $S \cap V(C)$ in $G[S]$
 - 12: **for** $v \in V(Co)$ **do**
 - 13: $score[v] \leftarrow score[v] + w[s] \cdot \frac{r_n(s) - w[S]}{|C|}$
 - 14: **end for**
 - 15: Contract $S \cup V(C)$ into s , deleting self-loops
 - 16: $w[s] \leftarrow w[S']$
 - 17: $r_n(s) \leftarrow r_n(s) - (|S \cup V(C)| - 1)$
 - 18: $r_m(s) \leftarrow r_m(s) - (|E[G[S]]| + |C|)$
 - 19: **end if**
 - 20: **until** $C = \emptyset$
 - 21: Uncontract all contracted vertices
 - 22: **end for**
 - 23: **return** the $kCNodes$ set that the cut-selection phase computes
-

Στη συνέχεια, κατά τη φάση Cut-selection (2.5), αναθέτονται βαθμολογίες σε όλες τις προσδιορισμένες τομές. Κάθε τομή C βαθμολογείται χρησιμοποιώντας τον

τύπο

$$score[C] = \sum_{(x,y) \in C} \max\{w[x_{in}] + w[x_{out}], w[y_{in}] + w[y_{out}]\}$$

Έπειτα, επιλέγεται η τομή με την υψηλότερη βαθμολογία και για κάθε μια από τις ακμές της, πραγματοποιείται διαγράφη αυτής με την υψηλότερη βαθμολογία. Είναι σημαντικό να σημειώσουμε ότι αυτή η διαδικασία μπορεί να αφαιρέσει τμήματα διαφορετικών μεγεθών (από 1 έως k). Παρ' όλα αυτά, σε κάθε περίπτωση, διαγράφεται το άκρο (κορυφή) που συνδέεται με την υψηλότερη βαθμολογία μεταξύ των άκρων κάθε τμήματος που τέμνεται.

Αλγόριθμος 2.5 Cut-selection

```
1: Initialize  $score(C) = 0$  for all  $C \in C$ 
2: for all  $C \in C$  do
3:   for all  $(x, y) \in C$  do
4:      $score(C) \leftarrow score(C) + \max\{score[x_{in}] + score[x_{out}], score[y_{in}] + score[y_{out}]\}$ 
5:   end for
6: end for
7: Let  $C_{max}$  be the cut with the maximum score among those in  $C$ 
8: Initialize  $kCNodes = \emptyset$ 
9: for all  $(x, y) \in C_{max}$  do
10:  if  $w[x_{in}] + w[x_{out}] > w[y_{in}] + w[y_{out}]$  then
11:     $kCNodes = kCNodes \cup \{x\}$ 
12:  else
13:     $kCNodes = kCNodes \cup \{y\}$ 
14:  end if
15: end for
16: return  $kCNodes$ 
```

2.3.3 Αλγόριθμος CutBased

CutBased χρησιμοποιεί απόλυτα τη μέθοδο που περιγράφεται στην ενότητα 2.3.2 μέχρι να διαγράψει nc κρίσιμους κόμβους (δείτε τον Αλγόριθμο 2.6). Αρχικά, εφαρμόζεται διαίρεση κορυφών στο G παράγοντας το G' με $n' = 2n$ κορυφές και $m' = m+n$ ακμές. Ο αλγόριθμος καλεί τη διαδικασία `kCriticalNodes` με δύο παραμέτρους: το n_{sample} , τον αριθμό των δειγματοληπτούμενων κορυφών, και το k , το μέγιστο μέ-

γεθος των τομών που λαμβάνονται υπόψη.

Για να διασφαλιστεί η αποτελεσματικότητα του αλγόριθμου, λαμβάνονται υπόψη δειγματοληπτικά σύνολα κορυφών με μεγάλο όγκο επικοινωνίας. Ωστόσο, η απλή δειγματοληψία μπορεί να μην είναι επαρκής, καθώς το γράφημα μειώνεται με την εξέλιξη του αλγόριθμου. Για να αντιμετωπιστεί αυτό το ζήτημα, προτείνεται η προσαρμογή του μεγέθους δειγματοληψίας βάσει μιας εκτίμησης rm του μέσου αριθμού προσβάσιμων ακμών. Η εκτίμηση rm υπολογίζεται με βάση τον αλγόριθμο του Cohen [19] και ενημερώνεται μετά από κάθε εκτέλεση του `kCriticalNodes`. Αυτή η προσέγγιση προσφέρει μια αξιόπιστη εκτίμηση του rm χωρίς επιπλέον υπολογιστικό κόστος. Το μέγεθος δειγματοληψίας (n_{sample}) ορίζεται ως $n_{\text{sample}} = \log(n') * \sqrt{\frac{m'}{rm}}$, το οποίο είναι μικρότερο από το μέγεθος που απαιτείται για γραμμικό αναμενόμενο χρόνο στον αλγόριθμο `kCriticalNodes`.

Επίσης, αυξάνεται συστηματικά το μέγεθος της τομής k . Αρχικοποιείται $k = 1$ και αυξάνεται με πιθανότητα να φτάσει στο $k = 10$, εξασφαλίζοντας ότι ο αλγόριθμος εξετάζει τομές μεγαλύτερου μεγέθους που μπορούν να επηρεάσουν σημαντικά το γράφημα. Ένας απλός ψευδοκώδικας αυτού του αλγορίθμου παρατίθεται στον αλγόριθμο 2.6.

Έχοντας μια ολοκληρωμένη ιδέα του αλγόριθμου, ενδιαφέρον αποτελεί ο χρόνος εκτέλεσης του. Ο χρόνος εκτέλεσης κυριαρχείται από τις κλήσεις του `kCriticalNodes`. Έστω $r = \frac{\sum_{s \in V'} rm(s)}{n'}$, όπου $rm(s)$ υποδηλώνει τον μέσο αριθμό ακμών που είναι προσβάσιμες από κάθε κορυφή s , και n' λειτουργεί ως παράγοντας κανονικοποίησης. Επιπρόσθετα, n_{sample} αναπαριστά τον αριθμό των δειγμάτων αρχικών κορυφών. Οπότε, ο αλγόριθμος λειτουργεί με χρονική πολυπλοκότητα $O(n_{\text{sample}} m' k^2)$ και απαιτεί χώρο $O(m' + n_{\text{sample}} n')$. Δεδομένου ότι $n_{\text{sample}} \leftarrow \log(n') \cdot \sqrt{\frac{m'}{rm}}$ τότε $O(m' r_m \log n)$ και $k = O(1)$, ο αλγόριθμος εκτελείται με αναμενόμενο χρόνο $O(m' \log n)$, χρησιμοποιώντας μνήμη $O(m' \log n)$ κατά μέσο όρο. Η ολοκληρωμένη απόδειξη βρίσκεται στο παραρτήμα Α.

Αλγόριθμος 2.6 CutBased

Require: $G, n_{\text{percentages}}$

- 1: Apply vertex-splitting to G producing G'
 - 2: $S \leftarrow \emptyset$
 - 3: **while** $|S| < nc$ **do**
 - 4: $k \leftarrow 1$
 - 5: **while** $\text{rand}([0, 1]) < \frac{1}{4}$ and $k < 10$ and $|S| + k + 1 \leq nc$ **do**
 - 6: $k \leftarrow k + 1$
 - 7: **end while**
 - 8: Estimate the average number rm of reachable edges
 - 9: $n_{\text{sample}} \leftarrow \log(n') \cdot \sqrt{\frac{m'}{rm}}$
 - 10: $S_k \leftarrow \text{kCriticalNodes}(G', n_{\text{sample}}, k)$ {Algorithm 2.4}
 - 11: Set $S \leftarrow S \cup S_k$ and delete S_k from G and G'
 - 12: Update m' to reflect the edge deletions
 - 13: **end while**
 - 14: Calculate the final transitive closure $T(G - S)$, where $S \subseteq V$ is the set of deleted critical nodes.
-

2.3.4 Αλγόριθμος CutBase-Degree

CutBase-Degree είναι ένας αλγόριθμος, όπως προδίδει και το όνομα του, βασισμένος στον αλγόριθμο CutBased (2.3.3). Η μόνη διαφορά είναι ο τρόπος που βαθμολογούν κάθε κορυφή. Ο συγκεκριμένος αλγόριθμος για κάθε βαθμολογία κορυφής πολλαπλασιάζει το βαθμό της.

$$\text{score}[C] = \sum_{(x,y) \in C} \max\{w[x_{in}] \cdot \text{degree}[x_{in}] + w[x_{out}] \cdot \text{degree}[x_{out}], w[y_{in}] \cdot \text{degree}[y_{in}] + w[y_{out}] \cdot \text{degree}[y_{out}]\}$$

Καθώς ο συγκεκριμένος αλγόριθμος είναι βασισμένος στον Cutbased και οι αλλαγές δεν επιβαρύνουν την πολυπλοκότητα, μπορούμε να πούμε με σιγουριά ότι ο χρόνος εκτέλεσης του Cutbased-Degree είναι $O(m' \log n)$.

2.3.5 Αλγόριθμοι CutBased1+ και CutBased2+

Οι δυο τελευταίοι αλγόριθμοι που θα οριστούν σε αυτήν την ενότητα είναι ο CutBased1+ και ο CutBased2+. Όπως τους προδίδει και το όνομα, αυτοί δυο αλγόριθμοι βασίζονται στον Cutbase (αλγόριθμο 2.6), απλά στη συγκεκριμένη περίπτωση υλοποιείται διαφορετικά ο αλγόριθμος kEdgeOut (2.3.1). Στην ουσία θα συνδυάσουμε τους αλγόριθμους Local1+ και Local2+ (από το άρθρο [2]) με τον αλγόριθμο kEdgeOut. Η βασική ιδέα είναι ο αλγόριθμος να δέχεται ως είσοδο μια κορυφή x ενός γραφήματος $G = (V, E)$ και δύο παραμέτρους ν και k έτσι ώστε $\nu k = O(|E|)$. Τα αποτελέσματα εξάγονται με τους ακόλουθους τρόπους:

- είτε επιστρέφει ένα σύνολο κορυφών S τέτοιο ώστε $x \in S$ και $|E(S, V \setminus S)| < k$
- είτε επιστρέφει \emptyset , πιστοποιώντας ότι δεν υπάρχει μη κενό σύνολο κορυφών S τέτοιο ώστε

$$x \in S, \text{vol}_{\text{out}}(S) \leq \nu, |E(S, V \setminus S)| < k \quad (2.1)$$

Εάν υπάρχει μη κενό σύνολο κορυφών S που ικανοποιεί την εξίσωση 2.1, τότε το \emptyset επιστρέφεται με πιθανότητα το πολύ $1/2$ (όπως προαναφέρθηκε 2.3.1)

Αυτό μπορούμε να το πετύχουμε αν γίνει πρόωρος τερματισμός στην DFS, που εκτελεί ο αλγόριθμος kEdgeOut (2.3). Με άλλα λόγια, ο αλγόριθμος DFS επαναλαμβάνεται για k φορές, αν τερματίσει κανονικά σε κάποια επανάληψη k' , όπου $k' < k$, τότε έχουμε βρει κάποιο S που ικανοποιεί τις συνθήκες. Αντιθέτως, αν δεν υπάρξει αυτός ο τερματισμός και οι επαναλήψεις ξεπεράσουν το k τότε θεωρούμε πως δεν υπάρχει σύνολο που ικανοποιεί τις συνθήκες. Η μόνη διαφορά μεταξύ του CutBased1+ και του CutBased2+ είναι ο τρόπος που θεσπίζουν το ποτέ θα γίνει ο πρόωρος τερματισμός και η επιλογή της κορυφής $y \in V(T)$ (όπου $V(T)$ είναι το σύνολο των κορυφών του DFS δέντρου T), με τέτοιο τρόπο ώστε ολόκληρος ο αλγόριθμος να βγάζει σωστά αποτελέσματα με σταθερή πιθανότητα.

Αλγόριθμος 2.7 kEdgeOutWithLocal1+(G, s, k, R)

```
1: for  $i = \log(2k)$  to  $\log(R)$  do
2:    $(S, C) \leftarrow \emptyset$ .
3:    $\nu \leftarrow \frac{2^i}{k}$ 
4:   for  $i = 1$  to  $k + 1$  do
5:     if  $i == k$  then
6:        $\tau \leftarrow 2 \cdot \nu \cdot k$ 
7:     else
8:        $\tau \leftarrow$  Random integer in  $[1, 2 \cdot \nu \cdot k]$ 
9:     end if
10:    Execute a DFS from  $s$  visiting exactly  $\tau$ -th edges. Let  $T$  be the resulting DFS
    tree.
11:    if less than  $\tau$ -th edges were visited then
12:      Let  $S$  be the set of visited vertices by the last DFS traversal, and let  $C$  be
      the set of outgoing edges from  $S$ .
13:      Restore all reversed edges.
14:    end if
15:    Track the last vertex added to  $T$  as  $y$ 
16:    Sample an edge  $(x, y)$  uniformly at random among the visited edges and
    reverse the path  $T[s, x]$ .
17:  end for
18:  if  $S \neq \emptyset$  then
19:    return  $(S, C)$ .
20:  end if
21:  Restore all reversed edges and  $(S, C) \leftarrow \emptyset$ .
22: end for
23: return  $(\emptyset, \emptyset)$ .
```

CutBased1+ σταματάει την ανάπτυξη του δέντρου DFS μόλις $\text{vol}_{\text{out}}(V(T)) \geq \tau$, όπου τ επιλέγεται τυχαία από το $[1, 2\nu k]$. Στην τελευταία επανάληψη, αντί αυτού, ορίστε $\tau \leftarrow 2\nu k$. Τέλος, ορίζεται το y να είναι η τελευταία κορυφή που προστέθηκε στο δέντρο DFS. Σημειώνεται ότι το vol_{out} μπορεί να υπολογιστεί ως το άθροισμα των βαθμών των επισκεπτόμενων κορυφών. Επίσης, μια ακμή είναι νέα εάν δεν έχει προσεγγιστεί σε προηγούμενες επαναλήψεις. Διαφορετικά, είναι παλιά. Ένας απλός

ψευδοκώδικας αυτού του αλγορίθμου παρατίθεται στον αλγόριθμο 2.7.

Αλγόριθμος 2.8 kEdgeOutWithLocal2+(G, s, k, R)

```

1: for each vertex  $v \in V$  do
2:   Initialize  $c(v)$  as the remaining capacity, initially  $c(v) = \deg_{\text{out}}(v)$ 
3: end for
4: for  $i = \log(2k)$  to  $\log(R)$  do
5:    $(S, C) \leftarrow \emptyset$ .
6:    $\nu \leftarrow \frac{2^i}{k}$ 
7:   for  $i = 1$  to  $k + 1$  do
8:     Set  $\tau \leftarrow$  random integer in  $[1, 2\nu]$ 
9:     Execute a DFS until  $\sum_{v \in V(T)} c(v) < \tau$ . Let  $T$  be the resulting DFS tree.
10:    Let  $u$  be the last visited vertex during DFS expansion
11:     $c(u) \leftarrow (\sum_{v \in T} c(v)) - \tau$ 
12:    for each vertex  $v \in T \setminus \{u\}$  do
13:       $c(v) \leftarrow 0$ 
14:    end for
15:    if less than  $\tau$ -th edges were visited then
16:      Let  $S$  be the set of visited vertices by the last DFS traversal, and let  $C$  be
        the set of outgoing edges from  $S$ .
17:      Restore all reversed edges.
18:    end if
19:    Track the last vertex added to  $T$  as  $y$ 
20:    Sample an edge  $(x, y)$  uniformly at random among the visited edges and
        reverse the path  $T[s, x]$ .
21:    end for
22:    if  $S \neq \emptyset$  then
23:      return  $(S, C)$ .
24:    end if
25:    Restore all reversed edges and  $(S, C) \leftarrow \emptyset$ .
26:  end for
27: return  $(\emptyset, \emptyset)$ .

```

CutBased2+ Για κάθε $v \in V$, έστω $c(v)$ η υπολειπόμενη χωρητικότητα για τον κόμβο v , που αντιπροσωπεύει τον αριθμό των μη καταμετρημένων εξερχόμενων

ακμών. Στην αρχή του, ορίζουμε $c(v) = \deg_{\text{out}}(v)$. Θέτει το $\tau \in [1, 2\nu]$ τυχαία πριν από το DFS. Ο αλγόριθμος, σταματάει να αναπτύσσει το δέντρο T αμέσως μόλις $\sum_{v \in V(T)} c(v) \geq \tau$. Παράλληλα, ενημερώνεται το $c(v)$ για κάθε επισκεπτόμενο κόμβο ως εξής:

- Για τον τελευταίο επισκεπτόμενο κόμβο u , θέτει $c(u) = (\sum_{v \in T} c(v)) - \tau$.
- Για τους άλλους κόμβους $v \in T$, θέτει $c(v) = 0$.

Έστω ότι T' είναι το δέντρο DFS στο σημείο όπου $\sum_{v \in V(T')} c(v) \geq \tau$ (πριν την ενημέρωση των χωρητικότητων). Σε αυτό το σημείο, θέτει το y ως τον τελευταίο κόμβο που προστέθηκε στο T' . Εν τέλει, μετριούνται προηγουμένες εξερχόμενες ακμές και επιλέγεται ο αρχικός κόμβος για μια από αυτές τυχαία. Ένας απλός ψευδοκώδικας αυτού του αλγορίθμου παρατίθεται στον αλγόριθμο 2.8.

Οι παραπάνω δυο αλγόριθμοι έχουν χρόνο εκτέλεσης $O(m' \log n)$. Μπορούμε να συμπεράνουμε κάτι τέτοιο καθώς και οι δυο αλγόριθμοι προέρχονται από τον CutBased αλγόριθμο και οι αλλαγές που γίνονται στον kEdgeOut δεν επηρεάζουν την συνολική πολυπλοκότητα.

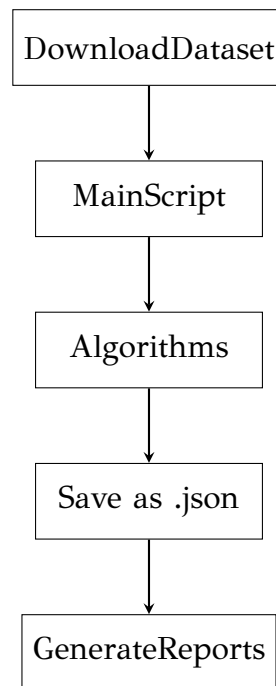
Έχοντας εξηγήσει και διατυπώσει όλους τους αλγόριθμους μπορούμε να αναπαράξουμε τον εξής πίνακα.

Αλγόριθμος	Τεχνική	Χρονος	Αναφορά
PageRank	select the vertex with the maximum PageRank value	$O(m)$	[11]
Label Propagation Direct	Partition vertex set into communities and select vertex with maximum number of neighbors in other communities. We let the algorithm iterate until the labels converge.	$O(m)$	[9]
Label Propagation Underacted	Apply LabelPropagation-D in the undirected version of the graphs (ignoring edge directions)	$O(m)$	[9]
CutBased	Return the $k = O(1)$ vertices selected by Algorithm 2.6	$O(m' \log n)$ expectation	
CutBased-Degree	Return the $k = O(1)$ vertices selected by the algorithm in 2.6 where the score of each vertex is multiplied by its degree	$O(m' \log n)$ expectation	
CutBased1+	Return the $k = O(1)$ vertices, stop DFS as soon as $\text{vol}_{\text{out}}(V(T)) \geq \tau$	$O(m' \log n)$ expectation	
CutBased2+	Return the $k = O(1)$ vertices, stop DFS as soon as $\sum_{v \in V(T)} c(v) \geq \tau$	$O(m' \log n)$ expectation	

Πίνακας 2.1: Μια επισκόπηση των αλγορίθμων που εξετάστηκαν. Τα όρια αναφέρονται με n κορυφές και m ακμές. Το αναφερόμενα όρια για τους LabelPropagation και PageRank υποθέτουν ότι εκτελούνται για σταθερό αριθμό επαναλήψεων

2.4 Οργάνωση πειραμάτων

Στην παρούσα ενότητα περιγράφονται λεπτομερώς οι προδιαγραφές του μηχανήματος που χρησιμοποιήθηκε για την εκτέλεση των δοκιμών, καθώς και τα αποτελέσματα των αλγορίθμων μας που υλοποιήθηκαν σε C++ χρησιμοποιώντας το g++ v.7.4.0 με πλήρη βελτιστοποίηση (flag -O3) για τη μεταγλώττιση του κώδικα. Οι χρόνοι εκτέλεσης μετρήθηκαν σε μηχάνημα GNU/Linux με Ubuntu (18.04.5 LTS): έναν διακομιστή Dell PowerEdge R715 64-bit NUMA με τέσσερις επεξεργαστές AMD Opteron 6376 και 128GB μνήμης RAM. Κάθε επεξεργαστής διαθέτει 8 πυρήνες που μοιράζονται 16MB L3 cache, και κάθε πυρήνας έχει 2MB ιδιωτική L2 cache και ταχύτητα 2300MHz. Στα πειράματά μας δεν χρησιμοποιήθηκε παράλληλη επεξεργασία, και κάθε αλγόριθμος εκτελέστηκε σε έναν μόνο πυρήνα. Οι χρόνοι CPU αναφέρονται μετρώντας με τη λειτουργία *high_resolution_clock* της βιβλιοθήκης *chrono* της *standardlibrary*, με μέσους όρους δέκα διαφορετικών εκτελέσεων.



Σχήμα 2.1: Ροή εργασίας πειραμάτων.

Αυτό το διάγραμμα αναπαριστά τη ροή εργασίας πειράματων μέσω μιας σειράς διαδοχικών βημάτων. Ξεκινά με τη διαδικασία "Λήψη Δεδομένων", όπου το σύνολο δεδομένων κατεβαίνει από το διαδίκτυο. Αυτά τα δεδομένα στη συνέχεια εισάγονται στο "Main Script" για αρχική επεξεργασία και προετοιμασία. Το "Main Script" διαχειρίζεται την εκτέλεση πολλαπλών αλγορίθμων (2.1), όπου πραγματοποιείται

η εύρεση κρίσιμων κορυφών και ο υπολογισμός $f(G)$ στο σύνολο δεδομένων. Τα αποτελέσματα αυτών των αλγορίθμων αποθηκεύονται ως αρχείο JSON. Τέλος, το αποθηκευμένο αρχείο JSON χρησιμοποιείται στο βήμα "Δημιουργία Αναφορών" για τη δημιουργία διαγραμμάτων και πινάκων σε μορφή Excel, συνοψίζοντας τα αποτελέσματα ολόκληρης της επεξεργαστικής διαδικασίας. Κάθε διαδικασία συνδέεται με βέλη που υποδεικνύουν τη ροή δεδομένων και ελέγχου από το ένα βήμα στο επόμενο, απεικονίζοντας την γραμμική πρόοδο από την απόκτηση δεδομένων μέχρι τη δημιουργία αναφορών.

ΚΕΦΑΛΑΙΟ 3

ΠΕΙΡΑΜΑΤΙΚΗ ΑΝΑΛΥΣΗ

3.1 Σύνολα δεδομένων

3.2 Πειραματική ανάλυση

Στα πειράματα που υλοποιήσαμε, συγκρίθηκαν συνολικά επτά αλγόριθμοι για τον υπολογισμό ενός κατά προσέγγιση πιο κρίσιμο κόμβο, οι οποίοι αναφέρονται αναλυτικά στον πίνακα 2.1. Όλοι οι αλγόριθμοι υλοποιήθηκαν με όσο το δυνατόν μεγαλύτερη αποδοτικότητα και ομοιομορφία, χρησιμοποιώντας ένα κοινό πλαίσιο (2.4). Στη συγκεκριμένη ενότητα παρουσιάζονται τα αποτελέσματα, όπου προέκυψε έπειτα από τον υπολογισμό της συνάρτησης συνδεσιμότητας $f(G \setminus S)$ έως ότου διαγραφεί ένα συγκεκριμένο ποσοστό των κορυφών, δηλαδή όταν $|S| = |V|$, για $0 < < 1$. Συγκεκριμένα, παρουσιάζονται αποτελέσματα στο εύρος $a = \{5, 10, 15, \dots, 45\}$.

3.1 Σύνολα δεδομένων

Τα γραφήματα που χρησιμοποιήθηκαν στα πειράματά μας, τα οποία προέρχονται από πραγματικά δεδομένα αναφέρονται στον πίνακα 3.1

Graphs	Type	SCC_s	$f(G)$	n	m	Ref
NotreDame	Web graph	15,389	2,373,102,510	48,715	267,647	[20]
Twitter	Social network	1,005	2,977,350	1,726	6,910	[15]
Google	Web Graph	406	242,041,770	15,763	152,945	[15]
Rome	Road network	1,978	11,239,256	3,353	8,870	[21]
Lastfm Asia	Social network	2,455	58,117,752	7,624	55,612	[20]
Food	Food Chains	26	13,193	128	2,106	
Gnutella25	P2p network	3,578	26,537,952	5,152	17,691	[20]

Πίνακας 3.1: Τα γραφήματα που χρησιμοποιήθηκαν στα πειράματα, τα n και m είναι οι αριθμός των κορυφών και των ακμών, αντίστοιχα, SCC_s είναι ο αριθμός των ισχυρά συνδεδεμένων συνιστωσών, και $f(G)$ είναι το μέγεθος του μεταβατικού κλεισίματος.

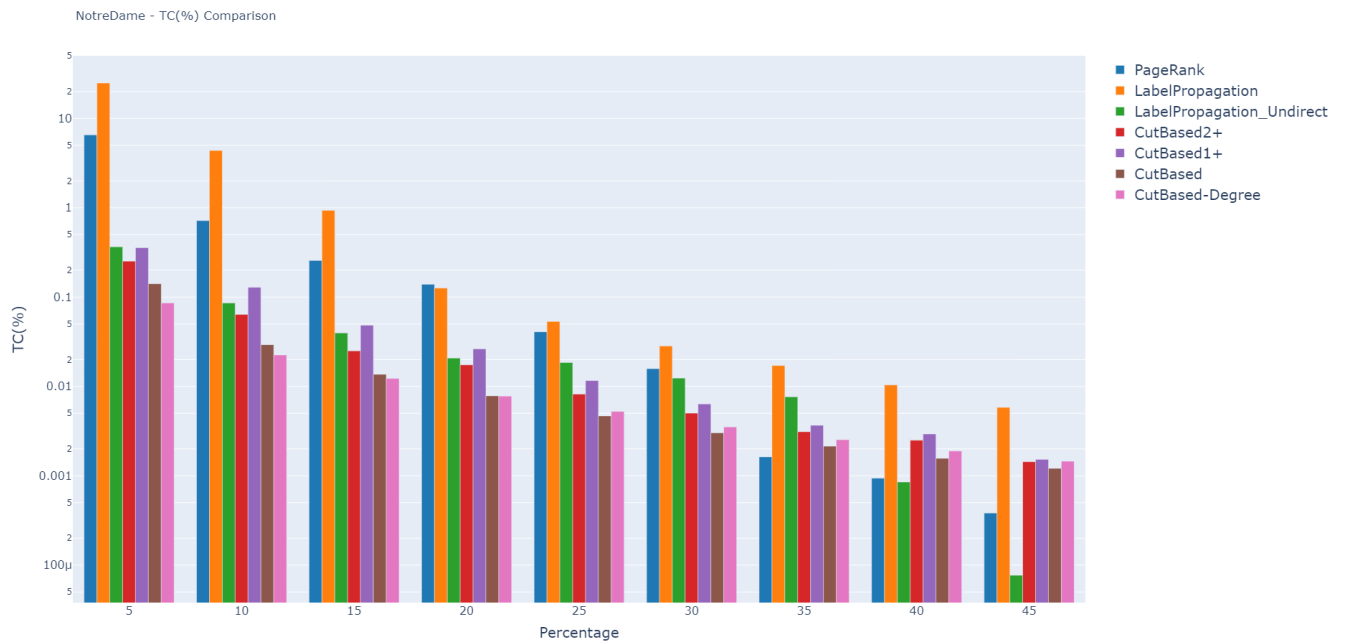
3.2 Πειραματική ανάλυση

Για να αποφύγουμε την απεικόνιση πολλών διαγραμμάτων, η οποία θα καθιστούσε δύσκολη την ανάγνωση, παρουσιάζουμε τη συμπεριφορά των αλγορίθμων σε πίνακες, για κάθε γράφημα που βρίσκεται στο παράρτημα Β. Αυτό το κεφάλαιο, περιλαμβάνει ανάλυση σε διαδικτυακά και κοινωνικά γραφήματα πραγματικών δεδομένων. Εμβαθύνοντας έτσι στην απόδοση του κάθε αλγορίθμου σε ένα συγκεκριμένο είδος γραφήματος.

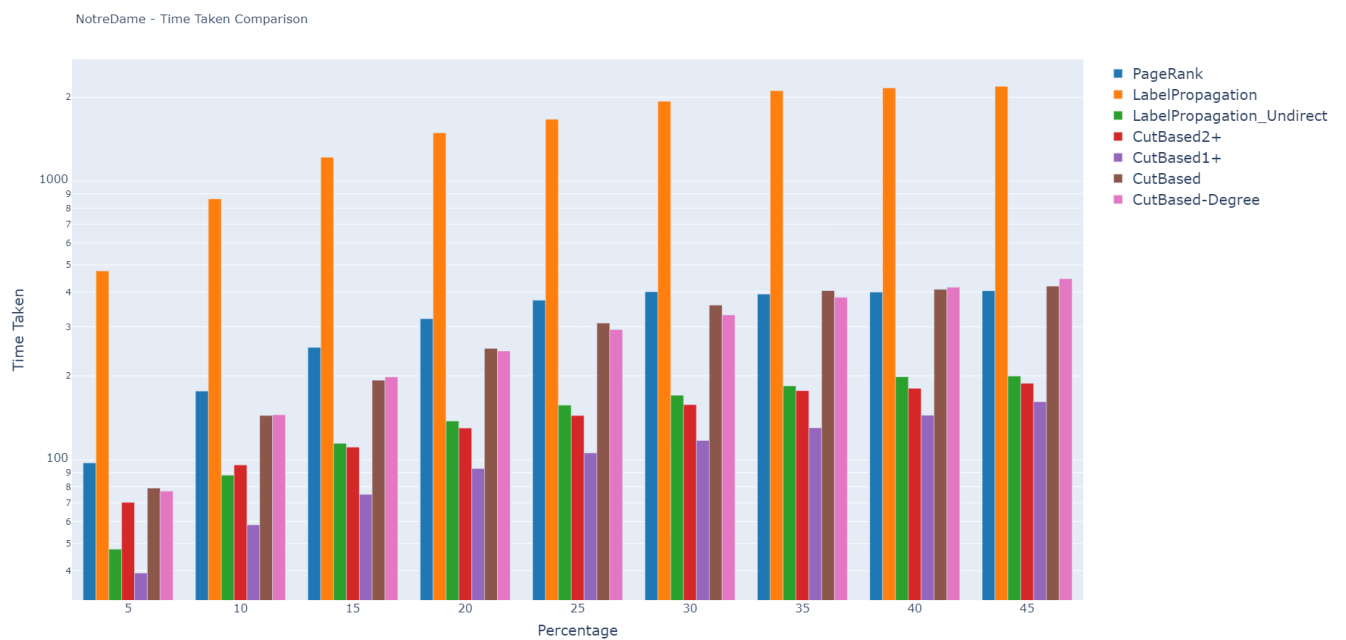
3.2.1 Διαδικτυακά γραφήματα

Στην κατηγορία διαδικτυακών γραφών θα περιμέναμε τόσο τον PageRank όσο και τον LabelPropagation (και τον LabelPropagation Underacted) να είναι αρκετά αποτελεσματικοί. Ωστόσο, από τα διαγράμματα παρατηρούμε ότι ο αριθμός των διαγραμμένων κορυφών είναι ίδιος, αλλά ο αριθμός των διαγραμμένων ακμών διαφέρει σημαντικά. Συγκεκριμένα, ο αλγόριθμος Label Propagation διαγράφει περισσότερες ακμές σε σύγκριση με άλλους αλγορίθμους, όπως φαίνεται αναλυτικά στο παράρτημα Β.1. Όσον αφορά τη συνάρτηση συνδεσιμότητας $f(G)$, που αντιπροσωπεύει τις εναπομείνουσες τιμές της αρχικής μεταβατική κλειστότητα του γραφήματος G , παρατηρούμε ότι μειώνεται δραματικά καθώς αυξάνεται το ποσοστό των διαγραμμένων κορυφών, γεγονός που είναι λογικό αφού αφαιρούνται κορυφές από το γράφημα

3.1α.



(α) $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές.

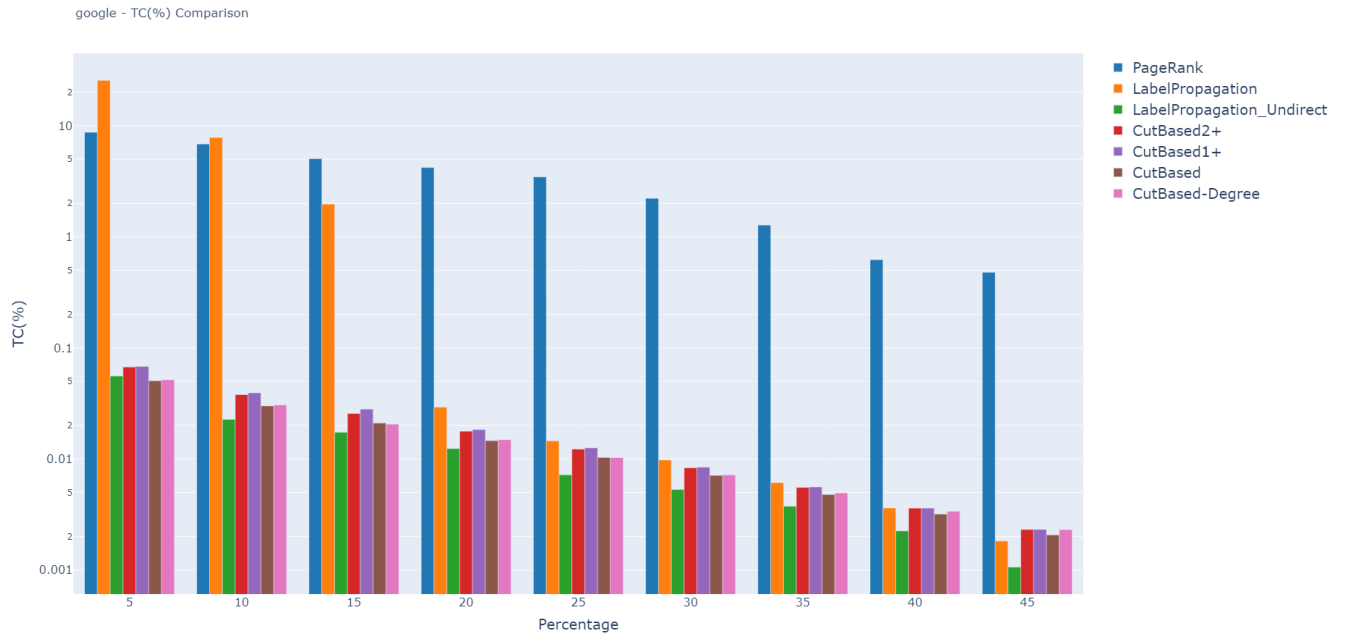


(β) Χρόνος εκτέλεσης αλγορίθμων.

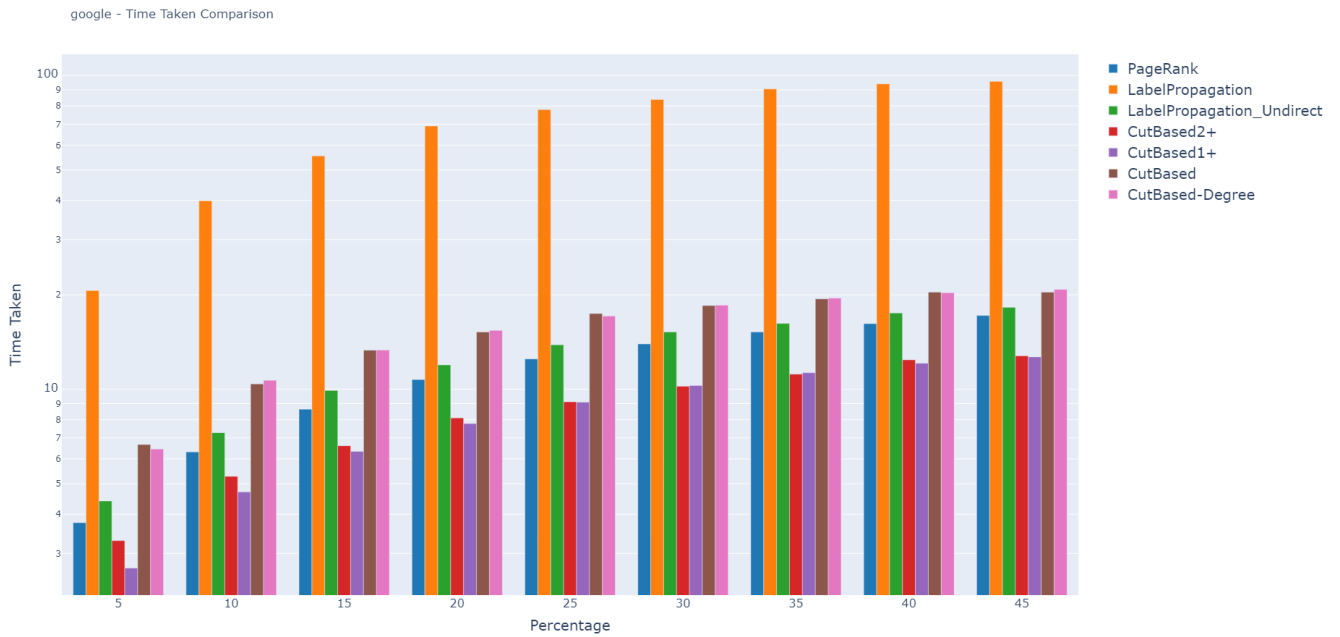
Σχήμα 3.1: NorthDame αποτέλεσμα

Οι αλγόριθμοι Label Propagation και PageRank διατηρούν υψηλότερες τιμές $f(G)$

σε χαμηλά ποσοστά διαγραφής κορυφών, ωστόσο σε μεγάλα ποσοστά ο αλγόριθμος LabelPropagation Underacted είναι ο πιο αποτελεσματικός από όλους. Επίσης παρατηρούμε ότι ο αλγόριθμος CutBased-Degree παραμένει σταθερά αποτελεσματικός σε χαμηλά ποσοστά, αλλά μετά το 25% ο αλγόριθμος CutBased έχει καλύτερα αποτελέσματα. Τέλος, οι αλγόριθμοι CutBased1+ και CutBased2+ δείχνουν να κρατάνε μια συνέπεια, καθώς έχουν καλύτερα αποτελέσματα από τους PageRank και Label-Propagation αλλά ελαφρώς χειρότερα από τους CutBased και CutBased-Degree, με τον CutBased2+ να είναι καλύτερος από τον CutBased1+. Όσον αφορά τον χρόνο εκτέλεσης (Time) 3.1β, παρατηρούμε ότι ο αλγόριθμος Label Propagation είναι ο πιο χρονοβόρος σε όλες τις περιπτώσεις, υποδεικνύοντας υψηλή πολυπλοκότητα. Όμως, ο Label Propagation Underacted εκτελείται πολύ πιο γρήγορα. Οι υπόλοιποι αλγόριθμοι εμφανίζουν χαμηλότερους χρόνους εκτέλεσης, με τους CutBased2+ και CutBased1+ να ξεχωρίζουν για την αποδοτικότητά τους.



(α) $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές.



(β) Χρόνος εκτέλεσης αλγορίθμων.

Σχήμα 3.2: Google αποτέλεσμα

Αντίστοιχα, στον δεύτερο γράφημα της ίδιας κατηγορίας (εικόνα 3.2), οι χρόνοι σύγκρισης είναι παρόμοιοι και επομένως δεν υπάρχει λόγος να τους σχολιάσουμε. Ωστόσο, στα αποτελέσματα της συνάρτησης συνδεσιμότητας $f(G)$ παρατηρούμε ότι

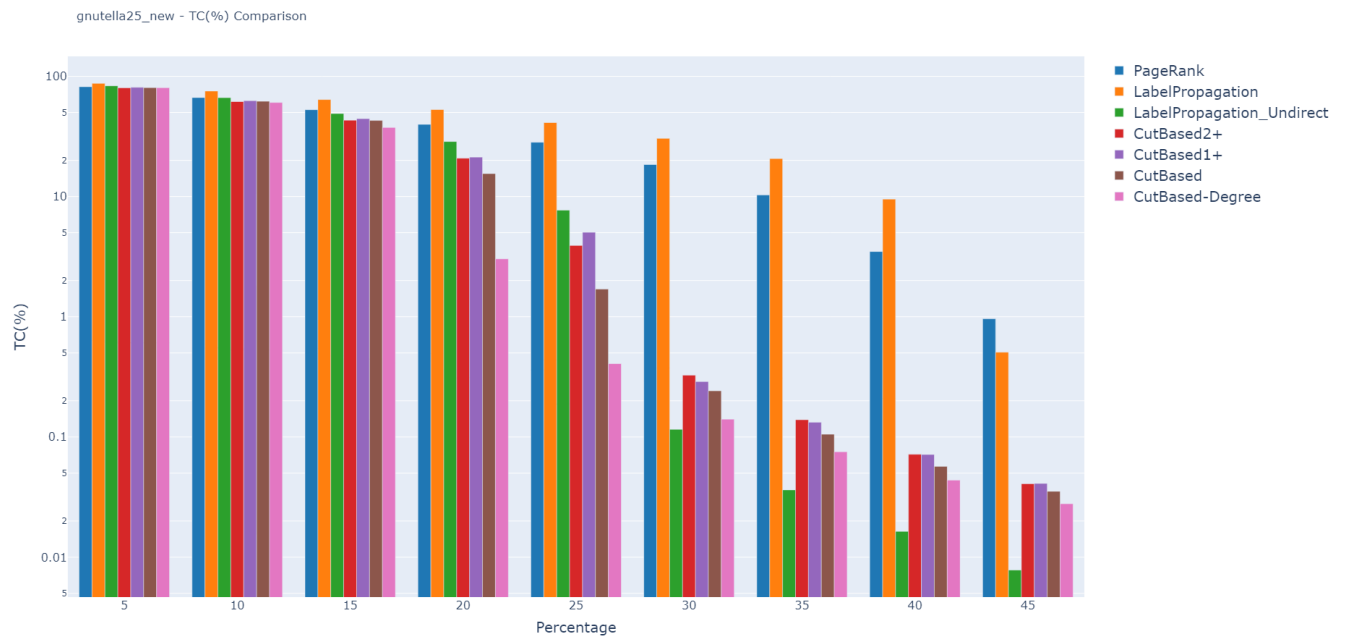
μετά το πείραμα με διαγραφή του 20% των κορυφών, ο Label Propagation παρουσιάζει αποτελέσματα παρόμοια με εκείνα των αλγορίθμων CutBased. Ειδικότερα, ο LabelPropagation Underacted είναι καλύτερος σε όλες τις περιπτώσεις. Όλα τα πειράματα φαίνονται αναλυτικά στο παραρτήμα Β.3.

Συνοψίζοντας, οι αλγόριθμοι Label Propagation και PageRank διατηρούν καλή συνδεσιμότητα, πράγμα που δεν επιθυμούμε ιδιαίτερα. Αντίθετα, οι CutBased2+ και CutBased1+ πετυχαίνουν καλή ισορροπία μεταξύ απόδοσης και χρόνου εκτέλεσης, ειδικά σε υψηλά ποσοστά διαγραφής. Για εφαρμογές όπου ο χρόνος εκτέλεσης είναι κρίσιμος, οι αλγόριθμοι CutBased-Degree και CutBased μπορεί να είναι προτιμητέροι από τους CutBased1+ και CutBased2+.

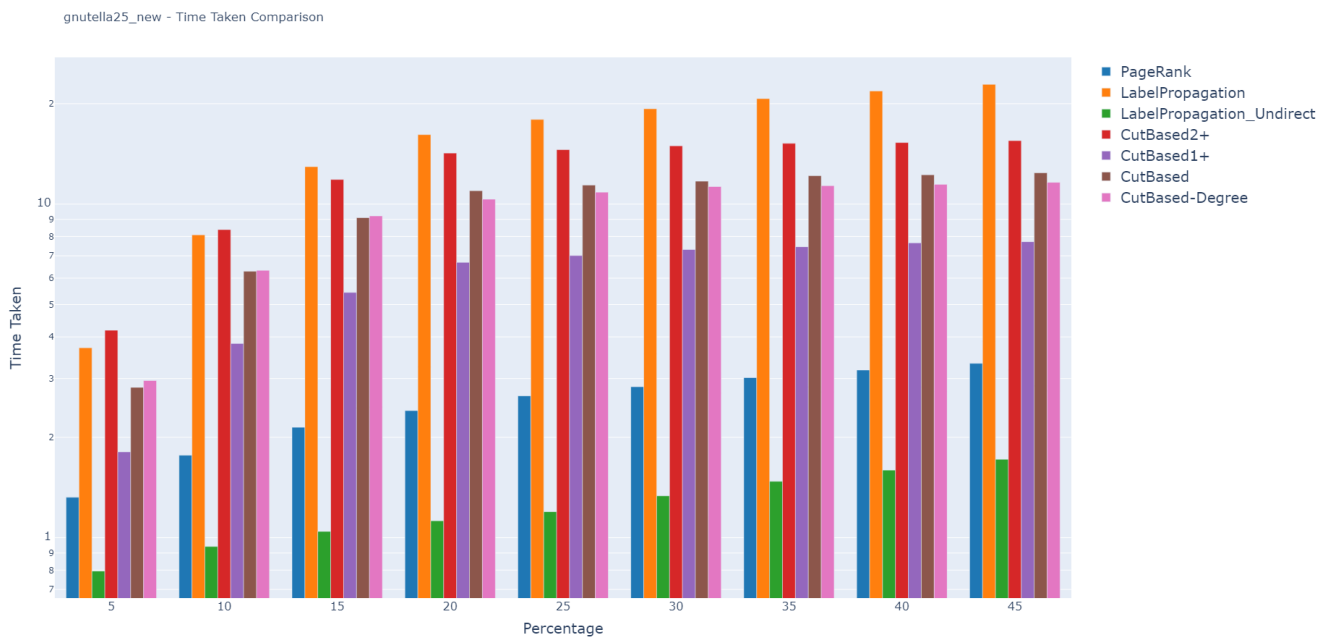
3.2.2 Γραφήματα πραγματικών δεδομένων

Τα δεδομένα παρουσιάζουν την απόδοση διάφορων αλγορίθμων στη διατήρηση της συνδεσιμότητας του δικτύου Gnutella25 (είκονα 3.3) μετά την αφαίρεση διαφορετικών ποσοστών των κορυφών του. Στόχος μας είναι να επιτύχουμε το μικρότερο ποσοστό της μεταβατική κλειστότητα (TC(%)) με βάση τον χρόνο εκτέλεσης (Time). Για την αφαίρεση 5% των κορυφών, ο αλγόριθμος CutBased-Degree επιτυγχάνει την μικρότερη τιμή του TC(%) στο 80.31% με χρόνο εκτέλεσης 2.96 δευτερόλεπτα. Στην αφαίρεση 10%, ο ίδιος αλγόριθμος διατηρεί τη χαμηλότερη τιμή TC(%) στο 60.53% με χρόνο εκτέλεσης 6.33 δευτερόλεπτα. Αυτά τα αποτελέσματα δείχνουν ότι ο CutBased-Degree επηρεάζει σημαντικά τη συνδεσιμότητα του δικτύου, αν και χρειάζεται περισσότερο χρόνο εκτέλεσης σε σύγκριση με άλλους αλγόριθμους. Καθώς αυξάνεται το ποσοστό αφαίρεσης κορυφών στο 15%, ο CutBased-Degree και πάλι διατηρεί τη χαμηλότερη τιμή TC(%) στο 37.56%, ενώ χρειάζεται 9.21 δευτερόλεπτα για την εκτέλεση. Στην αφαίρεση 20%, επιτυγχάνει το χαμηλότερο TC(%) στο 3.03% με χρόνο εκτέλεσης 10.35 δευτερόλεπτα. Στα επίπεδα αφαίρεσης 25%, 30%, 35%, και 40%, ο CutBased-Degree συνεχίζει να έχει τη χαμηλότερη τιμή TC(%) (0.41%, 0.14%, 0.08% και 0.04% αντίστοιχα), αν και οι χρόνοι εκτέλεσης κυμαίνονται από 10.85 έως 11.46 δευτερόλεπτα. Παράλληλα και οι αλγόριθμοι CutBased1+ και CutBased2+ εμφανίζουν ενδιαφέροντα αποτελέσματα. Καθώς αφαιρείται το 10% των κορυφών, ο CutBased2+ επιτυγχάνει TC(%) στο 61.62% με χρόνο εκτέλεσης 8.39 δευτερόλεπτα, ενώ ο CutBased1+ έχει ελαφρώς καλύτερο TC(%) στο 62.69% και χρόνο εκτέλεσης 3.82 δευτερόλεπτα. Αυτό υποδηλώνει ότι ο CutBased1+ είναι

πιο αποδοτικός σε χρόνο και προσφέρει καλύτερη διατήρηση της συνδεσιμότητας σε σχέση με τον CutBased2+ σε αυτό το επίπεδο αφαίρεσης.



(α) $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές.

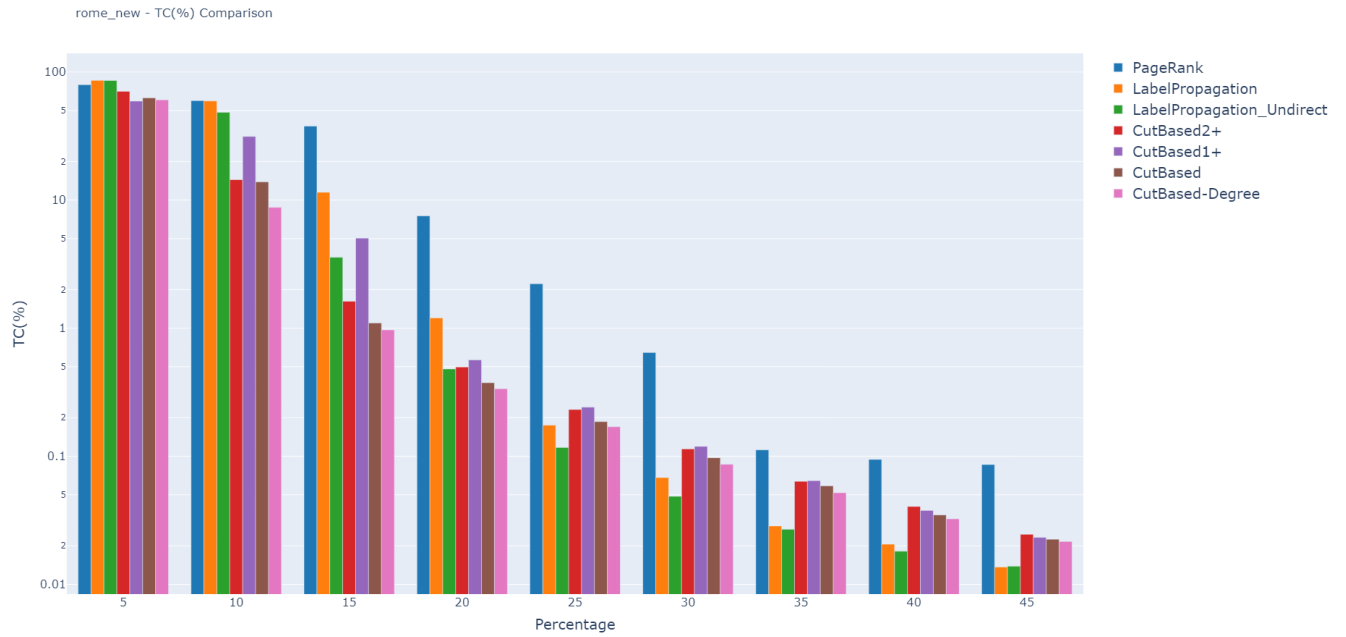


(β) Χρόνος εκτέλεσης αλγορίθμων.

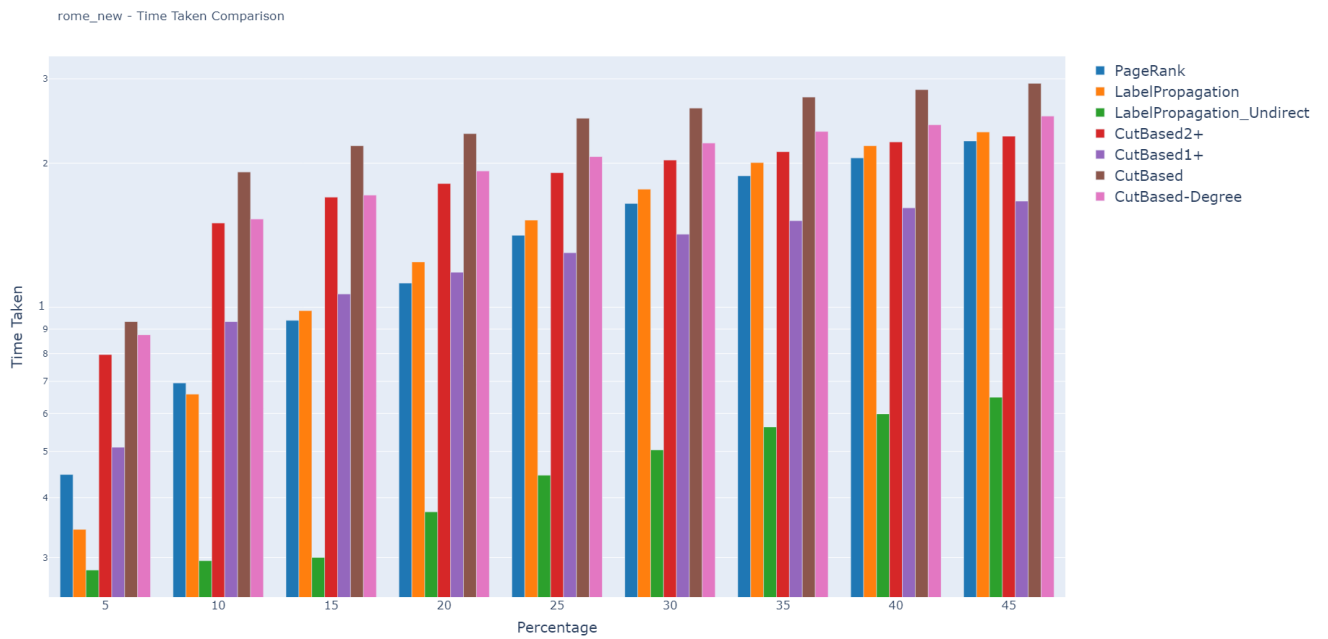
Σχήμα 3.3: Gnutella25 αποτέλεσμα

Στη συνέχεια αναλύονται δεδομένα του γραφήματος Rome, όπως φαίνεται στο

σχήμα (3.4). Ο χρόνος εκτέλεσης μεταξύ των αλγορίθμων διαφέρει αρκετά, με τις μεθόδους LabelPropagation γενικά να είναι ταχύτερες. Αυτή η αποτελεσματικότητα μπορεί να επωφεληθεί δίκτυα μεγάλης κλίμακας όπου οι υπολογιστικοί πόροι αποτελούν έναν σημαντικό παράγοντα. Τα αποτελέσματα που προέκυψαν, αντικατοπτρίζουν τις εγγενείς στρατηγικές αυτών των αλγορίθμων στην αφαίρεση κορυφών και τις επιπτώσεις τους στη συνολική δομή του γραφήματος. Οι προσεγγίσεις του PageRank και LabelPropagation δίνουν προτεραιότητα στη διατήρηση της συνδεσιμότητας, ενώ οι μέθοδοι βασισμένες σε CutBased εστιάζουν στη διαίρεση του δικτύου, οδηγώντας συχνά σε ταχεία μείωση της μεταβατικής κλειστότητας καθώς αφαιρούνται περισσότερες κορυφές. Γενικά σε αυτό το είδος γραφών, είδαμε δυο παρόμοιες συμπεριφορές από τους αλγόριθμους.



(α) $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές.



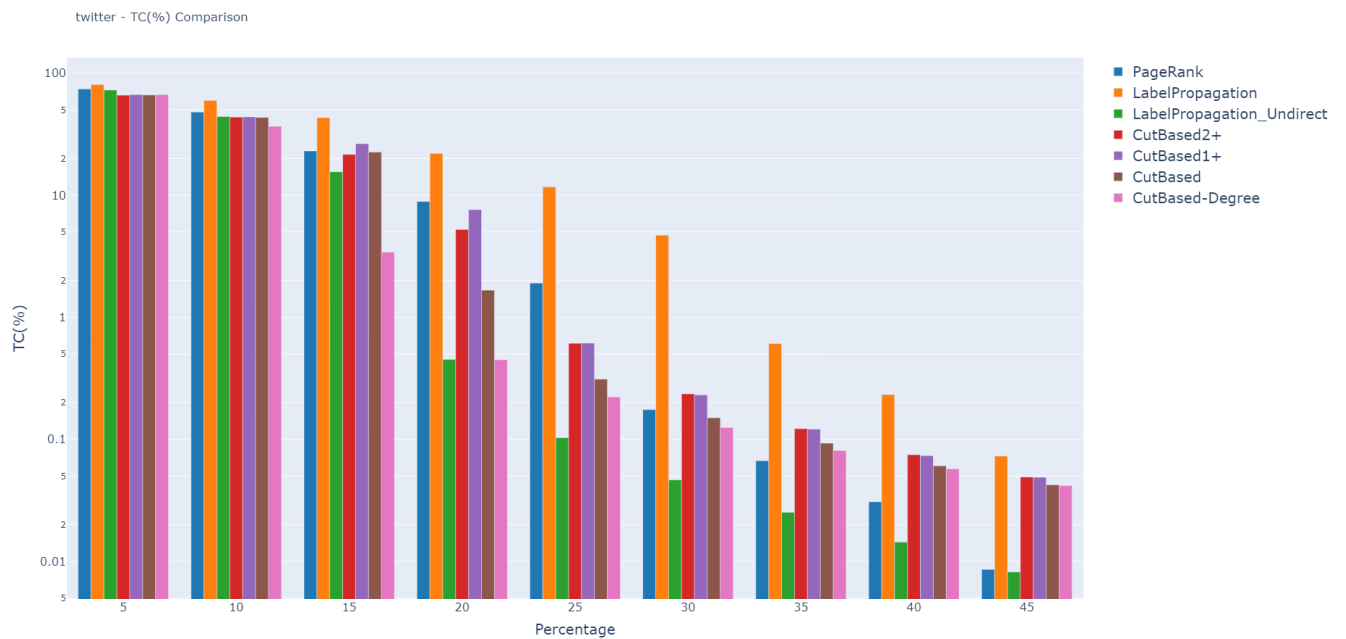
(β) Χρόνος εκτέλεσης αλγορίθμων.

Σχήμα 3.4: Rome αποτέλεσμα

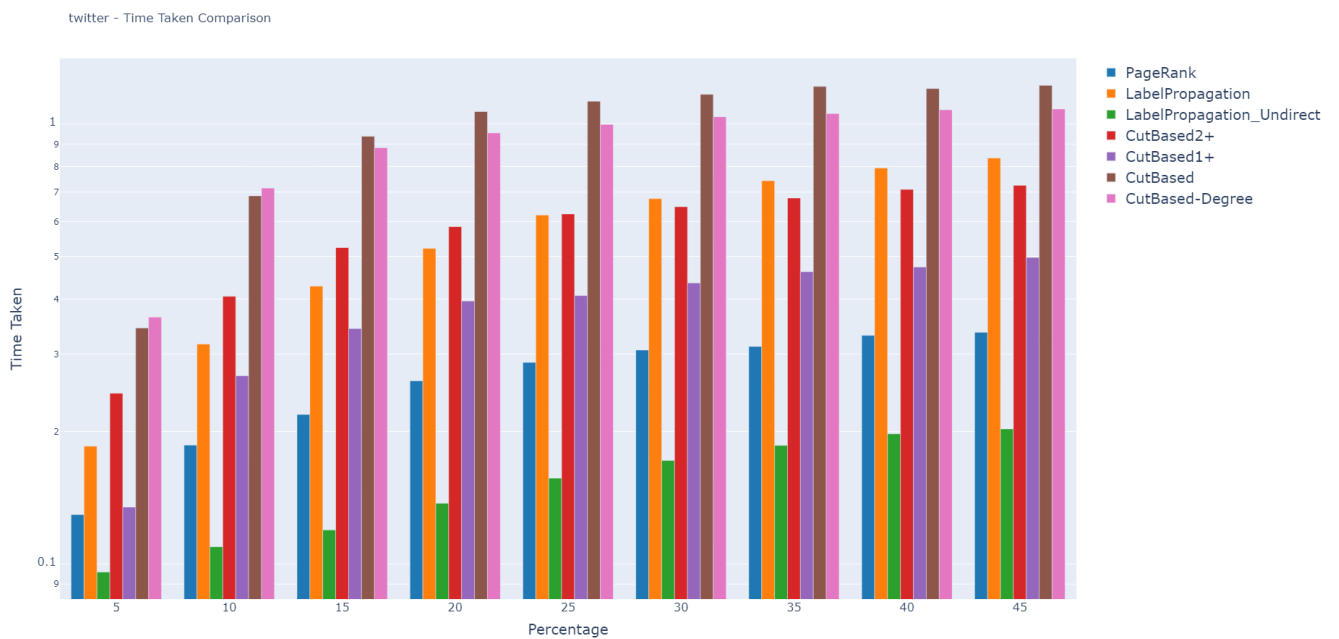
3.2.3 Γραφήματα κοινωνικών δικτύων

Τα αποτελέσματα των πειραμάτων στο σύνολο δεδομένων του Twitter (είκονα 3.5) αποκαλύπτουν ενδιαφέρουσες πτυχές σχετικά με την απόδοση διαφόρων αλγορίθμων. Αποδεικνύουν συγκεκριμένες τάσεις σε μείωση του TC(%) καθώς αυξάνεται το ποσοστό των αφαιρούμενων κορυφών από 5% έως 45%. Συγκεκριμένα, οι αλγόριθμοι PageRank και LabelPropagation εμφανίζουν συνήθως υψηλότερες τιμές TC(%) στα χαμηλά ποσοστά αφαίρεσης σε σύγκριση με τους CutBased και τις παραλλαγές τους, υποδηλώνοντας πιθανώς μια πιο επιλεκτική προσέγγιση στην αφαίρεση κορυφών που διατηρεί καλύτερα την αρχική δομή του γραφήματος. Ωστόσο, καθώς αφαιρούνται περισσότερες κορυφές, όλοι οι αλγόριθμοι συγκλίνουν σε χαμηλότερες τιμές TC(%), με τον CutBased-Degree να εμφανίζει σταθερά το χαμηλότερο TC(%) σε πολλά ποσοστά αφαίρεσης, αν και με το κόστος μεγαλύτερου χρόνου εκτέλεσης.

Η ανάλυση του συνόλου δεδομένων LastFm Asia (είκονα 3.6) ακολουθεί παρόμοια πρότυπα, αλγόριθμοι όπως ο PageRank και ο LabelPropagation εμφανίζουν αρχικά υψηλότερες τιμές TC(%) σε χαμηλά ποσοστά αφαίρεσης, υποδεικνύοντας μια πιο προσεκτική προσέγγιση στην αφαίρεση κορυφών σε σύγκριση με τις παραλλαγές CutBased. Καθώς αυξάνεται το ποσοστό αφαίρεσης, όλοι οι αλγόριθμοι εμφανίζουν σημαντική μείωση στο TC(%), υποδεικνύοντας μειωμένη συνδεσιμότητα του γραφήματος και μείωση της διατήρησης πληροφορίας. Ενδιαφέρον παρουσιάζει ο LabelPropagation Undirect, ο οποίος συστηματικά εμφανίζει υψηλότερες τιμές TC(%) σε σύγκριση με άλλους αλγορίθμους, υποδεικνύοντας την ικανότητά του να διατηρεί κατά κάποιο τρόπο τη δομή του γραφήματος. Παρόλα αυτά, το γεγονός αυτό συνδέεται από μεγαλύτερους χρόνους εκτέλεσης, ειδικά παρατηρούμενους σε αλγορίθμους όπως ο CutBased και ο CutBased-Degree, οι οποίοι επιτυγχάνουν χαμηλότερες τιμές TC(%) αλλά απαιτούν σημαντικά περισσότερους υπολογιστικούς πόρους, σε σχέση με τους CutBased1+ και CutBased2+.

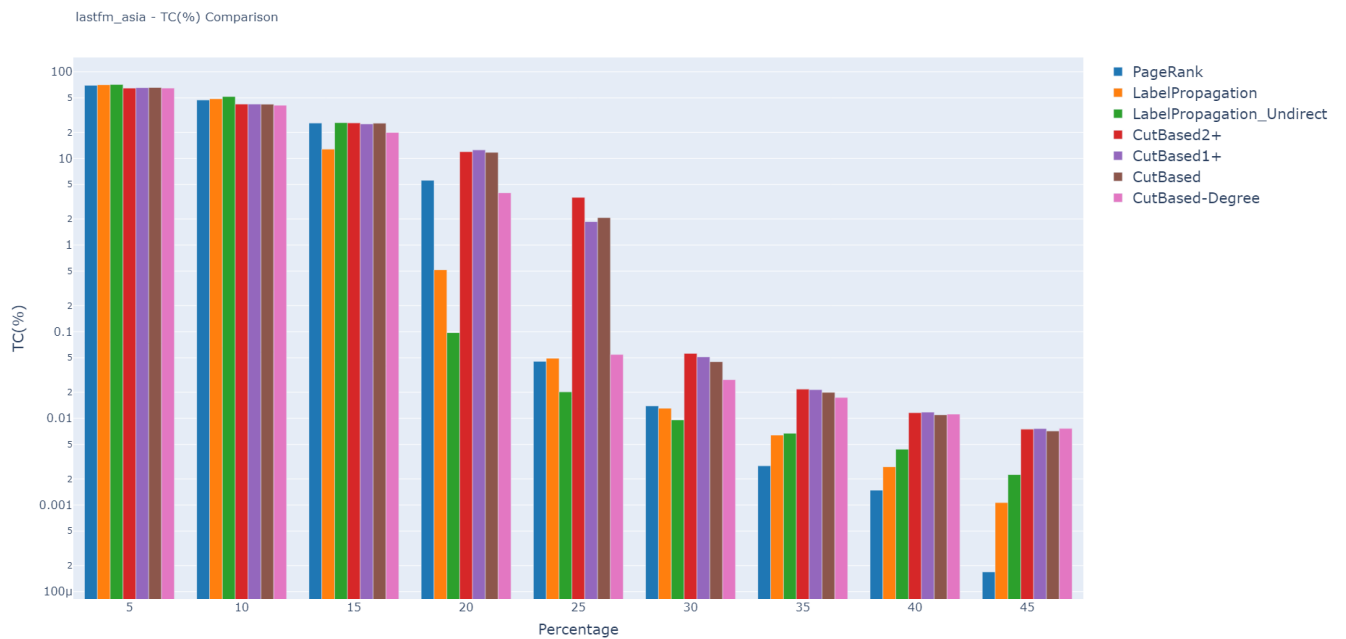


(α) $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές.

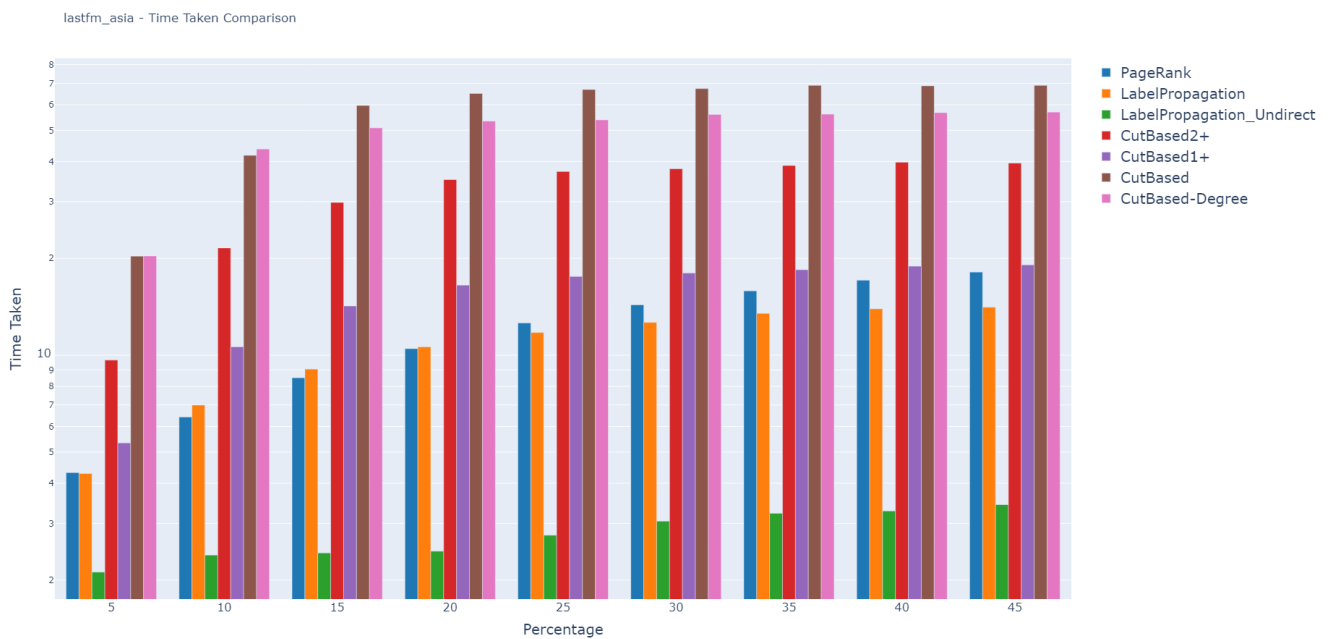


(β) Χρόνος εκτέλεσης αλγορίθμων.

Σχήμα 3.5: Twitter αποτέλεσμα



(α) $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές.



(β) Χρόνος εκτέλεσης αλγορίθμων.

Σχήμα 3.6: LastFm Asia αποτέλεσμα

ΚΕΦΑΛΑΙΟ 4

ΕΠΙΛΟΓΟΣ

Οι συγκρίσεις που υλοποιήθηκαν στη συγκεκριμένη εργασία αποδεικνύουν ότι ο CutBased-Degree είναι ο καλύτερος σύμφωνα με την συνάρτηση συνδεσιμότητας αλλά υστερεί πολύ σε χρόνο. Γενικά, παρατηρούμε ότι σε πολλά από τα πειράματά μας ο CutBased1+ και ο CutBased2+ είχαν περίπου ίδια, αν όχι ελαφρώς χειρότερη αποτελεσματικότητα από τους CutBased, CutBased-Degree, ωστόσο σε σημαντικά γρηγορότερο χρόνο. Από την άλλη, ο PageRank και ο LabelPropagation είναι πιο αποτελεσματικοί στη διατήρηση της δομής του δικτύου σε σύγκριση με τις μεθόδους CutBased. Αυτό οφείλεται στην προσέγγισή τους, να στοχεύουν κορυφές με βάση τη σημασία τους και της δομής της κοινότητας αντί για απλά κοψίματα του γραφήματος. Ίσως έχει μεγάλο ενδιαφέρον για μελλοντική έρευνα, οι παραλλαγές του CutBased1+ και ο CutBased2+ να συνδυαστούν με τον CutBased-Degree. Έτσι ώστε, να μπορέσουμε να αποκτήσουμε την ακρίβεια του CutBased-Degree όσο αφορά την συνάρτηση συνδεσιμότητα αλλά σε χρόνους όπως αυτών των CutBased1+ και CutBased2+.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] L. Georgiadis, G. F. Italiano, A. Karanasiou, and N. Parotsidis, “New algorithms for critical node detection in directed networks.”
- [2] M. Franck and S. Yingchareonthawornchai, “Engineering nearly linear-time algorithms for small vertex connectivity,” *Association for Computing Machinery*, vol. 27, no. 4, pp. 1–29, 2022.
- [3] D. Kempe, J. Kleinberg, and E. Tardos, “Maximizing the spread of influence through a social network.” *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, no. 3, pp. 137–146, 2003.
- [4] N. Paudel, L. Georgiadis, and G. F. Italiano, “Computing critical nodes in directed graphs,” *ACM Journal of Experimental Algorithmics*, vol. 23, no. 2, pp. 1–24, 2017.
- [5] S. P. Borgatti, “Centrality and network flow.” *Social Networks*, vol. 27, no. 1, pp. 55–71, 2005.
- [6] L. Tian, A. Bashan, D.-N. Shi, and Y.-Y. Liu, “Articulation points in complex networks.” *Nature communications*, vol. 8, no. 1, 2017. [Online]. Available: <https://doi.org/10.1038/ncomms14223>
- [7] A. Arulselvan, C. W. Commander, L. Elefteriadou, and P. M. Pardalos., “Detecting critical nodes in sparse graphs.” *Comput. Oper. Res.*, vol. 36, no. 7, pp. 2193–2200, 2009.
- [8] M. Lalou, M. A. Tahraoui, and H. Kheddouci., “The critical node detection problem in networks:a survey,” *Computer Science Review*, vol. 28, pp. 92–117, 2018. [Online]. Available: <https://doi.org/10.1016/j.cosrev.2018.02.002>
- [9] P. Boldi, M. Rosa, and S. Vigna, “Robustness of social and web graphs to node removal,” *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 829–842, 2013.

- [10] M. Barthélemy, “Betweenness centrality in large complex networks,” *The European Physical Journal B*, vol. 38, no. 2, pp. 163–168, 2014.
- [11] L. Page, S. Brin, R. Motwani, and T. Winograd., “The pagerank citation ranking: Bringing order to the web,” *Stanford InfoLab*, 1999. [Online]. Available: <http://ilpubs.stanford.edu:8090/422/>
- [12] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” *Physical Review E*, vol. 76, no. 3, pp. 1–11, 2007.
- [13] B. Florentine, H. W. Hamacher, F. Maffioli, and A. M. Schwahn, “Minimum cut bases in undirected networks,” *Discrete Applied Mathematics*, vol. 158, no. 4, pp. 177–290, 2010. [Online]. Available: <https://doi.org/10.1016/j.dam.2009.07.015>
- [14] S. Forster, D. Nanongkai, L. Yang, T. Saranurak, and S. Yingchareonthawornchai, “Computing and testing small connectivity in near-linear time and queries via fast local cut algorithms,” *In Proceeding of the SODA ,SIAM*, pp. 2046–2065, 2020. [Online]. Available: <https://doi.org/10.1137/1.9781611975994.126>
- [15] R. A. Rossi and N. K. Ahmed, “The network data repository with interactive graph analytics and visualization,” *AAAI*, 2015. [Online]. Available: <http://networkrepository.com>
- [16] S. Chechik, T. D. Hansen, G. F. Italiano, V. Loitzenbauer, and N. Parotsidis, “Faster algorithms for computing maximal 2-connected subgraphs in sparse directed graphs.” *In Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on 444 Discrete Algorithms*, p. 1900–1918, 2017. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3039686.3039810>.
- [17] D. Nanongkai, T. Saranurak, and S. Yingchareonthawornchai, “Computing and testing small vertex connectivity in near-linear time and querie,” 2019. [Online]. Available: <http://arxiv.org/abs/1905.05329>
- [18] S. Forster and L. Yang, “A faster local algorithm for detecting bounded-size cuts with applications to higher-connectivity problems,” 2019. [Online]. Available: <http://arxiv.org/abs/1904.08382>

- [19] E. Cohen, “Size-estimation framework with applications to transitive closure and reachability,” *Journal of Computer and System Science*, vol. 55, no. 3, p. 441 – 453, 1997. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0022000097915348>
- [20] J. Leskovec and A. Krevl, “Snap datasets: Stanford large network dataset collection,” 2014. [Online]. Available: <http://snap.stanford.edu/data>
- [21] C. Demetrescu, A. Goldberg, and D. Johnson, “9th dimacs implementation challenge: Shortest paths,” 2007. [Online]. Available: <http://www.dis.uniroma1.it/~challenge9/>

ΠΑΡΑΡΤΗΜΑ Α

ΑΠΟΔΕΙΞΗ ΧΡΟΝΟΥ ΕΚΤΕΛΕΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ CUTBASED

Στη φάση Cut-exploration, πρώτα υπολογίζεται η πληθικότητα των συνόλων των προσβάσιμων κορυφών και ακμών από μια δειγματοληπτημένη κορυφή s , αυτό απαιτεί χρόνο $O(rm(s))$. Τώρα εξετάζουμε τον υπόλοιπο χρόνο που απαιτείται για την συγκεκριμένη φάση. Πρώτα ας αναλύσουμε τον συνολικό χρόνο που δαπανάται σε όλες τις κλήσεις στην $kEdgeOut$. Κάθε επιτυχημένη κλήση στην $kEdgeOut$ επιστρέφει μια συνιστώσα $\langle S, C \rangle$ με μέγεθος $|S|$ σε χρόνο $O(|S|k^2)$, και οι ακμές από $G[S \cup V(C)]$ αφαιρούνται από το G' . Δεδομένου ότι συνολικά εξερευνείται το πολύ $O(rm(s))$ ακμές, ο συνολικός χρόνος που δαπανάται σε επιτυχημένες κλήσεις στην $kEdgeOut$ είναι $O(rm(s)k^2)$. Μια ανεπιτυχής κλήση στην $kEdgeOut$ απαιτεί χρόνο $O(rm(s)k)$, σύμφωνα με την ανάλυση της $kEdgeOut$ (2.3.1). Ο χρόνος που απαιτείται για όλες τις διασχίσεις του γραφήματος σε μια μοναδική κλήση στη φάση Cut-exploration είναι $O(rm(s))$, καθώς κάθε εξερεύνηση απαιτεί γραμμικό χρόνο στον όγκο του k -edge-out $\langle S, C \rangle$, και η συνιστώσα στη συνέχεια συστέλλεται, εξασφαλίζοντας ότι οι ακμές του δεν επισκέπτονται ξανά. Τέλος, οι υπόλοιπες λειτουργίες στον κύριο βρόχο, απαιτούν γραμμικό χρόνο στο μέγεθος του k -edge-out $\langle S, C \rangle$ και συνεπώς απαιτούν συνολικά χρόνο $O(rn(s)) = O(rm(s))$. Να σημειωθεί, ότι αυτό επίσης αποδεικνύει ότι ο συνολικός αριθμός των κορυφών που εισάγονται στη συλλογή των τομών C είναι $O(rn(s)) = O(rm(s))$ για κάθε εκτέλεση της φάσης Cut-exploration, που είναι $O\left(\sum_{i=1}^{n_{sample}} rm(s_i)\right)$ συνολικά, όπου s_i είναι η δειγματοληπτημένη κορυφή

στην i -η κλήση της φάσης Cut-exploration. Η φάση Cut-selection απαιτεί χρόνο γραμμικό στο μέγεθος του C , που είναι $O(\sum_{i=1}^{n_{sample}} rm(s_i))$ όπως μόλις αναφέραμε. Ο χώρος που απαιτείται από τον αλγόριθμό μας είναι $O(m' + \sum_{i=1}^{n_{sample}} rm(s_i))$, καθώς οι πρόσθετες δομές δεδομένων απαιτούν μόνο χώρο $O(1)$ ανά κορυφή, άρα συνολικά $O(n')$. Λαμβάνοντας υπόψη τον αριθμό των ακμών που είναι προσβάσιμες από κάθε δειγματοληπτημένη κορυφή, ο συνολικός χρόνος εκτέλεσης του αλγορίθμου είναι $O(\sum_{i=1}^{n_{sample}} rm(s_i)k^2)$ και ο χώρος που απαιτείται από τον αλγόριθμό μας είναι $O(\sum_{i=1}^{n_{sample}} rm(s_i))$. Τέλος, κάθε δειγματοληπτημένη κορυφή μπορεί να φτάσει $O(\sum_{s \in V} \frac{rm(s)}{n'}) = rm$ ακμές. Με τη γραμμικότητα της προσδοκίας, $O(\sum_{i=1}^{n_{sample}} rm(s_i)) = O(n_{sample}rm)$. Δεδομένου ότι $n_{sample} = O(\frac{m'}{rm} \log n)$, έχουμε $O(\sum_{i=1}^{n_{sample}} rm(s_i)) = O(m' \log n)$, το οποίο ολοκληρώνει την απόδειξη.

ΠΑΡΑΡΤΗΜΑ Β

ΑΠΟΤΕΛΕΣΜΑ ΓΙΑ ΟΛΑ ΤΑ ΠΕΙΡΑΜΑΤΑ

B.1 Πίνακες για NotreDame

B.2 Πίνακες για Food

B.3 Πίνακες για Google

B.4 Πίνακες για LastFm Asia

B.5 Πίνακες για Rome

B.6 Πίνακες για Twitter

B.7 Πίνακες για Gnutella25

B.1 Πίνακες για NotreDame

Πίνακας B.1: Αποτελέσματα της εκτέλεσης των αλγορίθμων για την αφαίρεση διαφορετικών ποσοστών κορυφών του NotreDame. Το $f(G)$ αναφέρεται στις υπόλοιπες τιμές της αρχικής τιμής transitive closure του G . $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές, και Time είναι ο χρόνος εκτέλεσης σε δευτερόλεπτα.

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
Removing 5% of the vertices					
PageRank	2436	29976	155015585	97.453655	6.532191

Πίνακας Β.1: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
LabelPropagation	2436	88075	589479927	476.140192	24.840053
LabelPropagation	2436	136956	8673151	47.749422	0.365477
Undirect					
CutBased2+	2436	40243	5984131	70.276034	0.252165
CutBased1+	2436	34134	8459570	39.217390	0.356477
CutBased	2436	39295	3342584	79.172214	0.140853
CutBased-	2436	54923	2038019	77.100951	0.085880
Degree					
Removing 10% of the vertices					
PageRank	4872	38923	16992606	176.323749	0.716050
LabelPropagation	4872	121376	103757820	863.181076	4.372243
LabelPropagation	4872	169324	2035306	88.056356	0.085766
Undirect					
CutBased2+	4872	72094	1517925	95.791326	0.063964
CutBased1+	4872	67782	3050657	58.406485	0.128551
CutBased	4872	76979	695463	144.184830	0.029306
CutBased-	4872	81208	533231	144.955137	0.022470
Degree					
Removing 15% of the vertices					
PageRank	7308	45789	6085217	253.088559	0.256425
LabelPropagation	7308	143687	22174613	1217.812324	0.934414
LabelPropagation	7308	190480	941630	114.475674	0.039679
Undirect					
CutBased2+	7308	100916	593757	110.943302	0.025020
CutBased1+	7308	95288	1148537	75.146140	0.048398
CutBased	7308	103513	324291	192.889362	0.013665
CutBased-	7308	102179	290971	198.299848	0.012261
Degree					
Removing 20% of the vertices					
PageRank	9743	52001	3297367	320.880347	0.138948
LabelPropagation	9743	165939	2993708	1490.234314	0.126152

Πίνακας Β.1: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
LabelPropagation	9743	206185	491859	137.714141	0.020726
Undirect					
CutBased2+	9743	114698	413290	129.948523	0.017416
CutBased1+	9743	115186	624996	93.004907	0.026337
CutBased	9743	112484	186035	250.889601	0.007839
CutBased-	9743	119764	184599	245.621348	0.007779
Degree					
Removing 25% of the vertices					
PageRank	12179	57446	969452	373.414046	0.040852
LabelPropagation	12179	182236	1265981	1665.298090	0.053347
LabelPropagation	12179	217747	438973	157.070950	0.018498
Undirect					
CutBased2+	12179	146687	194817	144.073645	0.008209
CutBased1+	12179	144344	275728	105.824054	0.011619
CutBased	12179	149946	110803	309.389544	0.004669
CutBased-	12179	136591	124464	293.274312	0.005245
Degree					
Removing 30% of the vertices					
PageRank	14615	64019	374384	400.579991	0.015776
LabelPropagation	14615	197606	670832	1933.604070	0.028268
LabelPropagation	14615	235053	294048	170.335294	0.012391
Undirect					
CutBased2+	14615	168989	119154	157.621892	0.005021
CutBased1+	14615	166878	151213	117.242360	0.006372
CutBased	14615	172974	71724	358.724582	0.003022
CutBased-	14615	154551	83476	330.842394	0.003518
Degree					
Removing 35% of the vertices					
PageRank	17051	67786	38619	392.858046	0.001627
LabelPropagation	17051	211521	406432	2110.266398	0.017127

Πίνακας Β.1: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
LabelPropagation	17051	246812	181390	184.069200	0.007644
Undirect					
CutBased2+	17051	190971	73862	176.950146	0.003112
CutBased1+	17051	188842	87153	130.283431	0.003673
CutBased	17051	193648	50804	404.354418	0.002141
CutBased-	17051	175155	60132	382.716733	0.002534
Degree					
Removing 40% of the vertices					
PageRank	19486	69525	22290	399.786634	0.000939
LabelPropagation	19486	223245	246349	2158.339343	0.010381
LabelPropagation	19486	258815	20174	198.526892	0.000850
Undirect					
CutBased2+	19486	190971	59221	180.414710	0.002496
CutBased1+	19486	202646	69597	144.456813	0.002933
CutBased	19486	206035	37146	408.843003	0.001565
CutBased-	19486	196740	44866	415.878313	0.001891
Degree					
Removing 45% of the vertices					
PageRank	21922	70783	9072	404.105831	0.000382
LabelPropagation	21922	234621	138130	2186.272940	0.005821
LabelPropagation	21922	266256	1827	199.800791	0.000077
Undirect					
CutBased2+	21922	226680	34044	188.019381	0.001435
CutBased1+	21922	226800	36117	161.418912	0.001522
CutBased	21922	227859	28688	419.627297	0.001209
CutBased-	21922	215006	34497	446.223568	0.001454
Degree					

B.2 Πίνακες για Food

Πίνακας B.2: Αποτελέσματα της εκτέλεσης των αλγορίθμων για την αφαίρεση διαφορετικών ποσοστών κορυφών του Food. Το $f(G)$ αναφέρεται στις υπόλοιπες τιμές της αρχικής τιμής transitive closure του G . $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές, και Time είναι ο χρόνος εκτέλεσης σε δευτερόλεπτα.

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
Removing 5% of the vertices					
PageRank	7	12	6975	0.001559	52.868946
LabelPropagation	7	169	12171	0.002611	92.253468
LabelPropagation	7	144	11966	0.002733	90.699613
Undirect					
CutBased2+	7	166	5104	0.000527	38.687183
CutBased1+	7	165	5235	0.001936	39.680133
CutBased	7	109	6403	0.002316	48.533313
CutBased-	7	163	6786	0.000000	51.436368
Degree					
Removing 10% of the vertices					
PageRank	13	31	4907	0.001016	37.193966
LabelPropagation	13	291	11088	0.005163	84.044569
LabelPropagation	13	370	9528	0.001507	72.220117
Undirect					
CutBased2+	13	253	4021	0.003244	30.478284
CutBased1+	13	272	4653	0.003189	35.268703
CutBased	13	220	4362	0.001359	33.062988
CutBased-	13	387	4413	0.003627	33.449557
Degree					
Removing 15% of the vertices					
PageRank	20	34	4334	0.001403	32.850754
LabelPropagation	20	390	10391	0.008149	78.761464

Πίνακας B.2: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
LabelPropagation Undirect	20	573	8577	0.002148	65.011749
CutBased2+	20	425	3517	0.004907	26.658076
CutBased1+	20	438	3882	0.002426	29.424695
CutBased	20	390	3661	0.001927	27.749564
CutBased- Degree	20	601	3843	0.010221	29.129084
Removing 20% of the vertices					
PageRank	26	59	3450	0.002308	26.150231
LabelPropagation	26	552	8433	0.009491	63.920261
LabelPropagation Undirect	26	732	7592	0.002844	57.545668
CutBased2+	26	566	2897	0.006438	21.958614
CutBased1+	26	612	3367	0.006430	25.521110
CutBased	26	495	3139	0.008637	23.792920
CutBased- Degree	26	771	3150	0.012379	23.876298
Removing 25% of the vertices					
PageRank	32	72	2963	0.002671	22.458880
LabelPropagation	32	715	7549	0.009354	57.219738
LabelPropagation Undirect	32	844	6679	0.004402	50.625332
CutBased2+	32	742	2062	0.006634	15.629500
CutBased1+	32	708	2766	0.006977	20.965664
CutBased	32	892	2791	0.012508	21.155158
CutBased- Degree	32	1000	2554	0.012704	19.358751
Removing 30% of the vertices					
PageRank	39	72	2594	0.004198	19.661942

Πίνακας B.2: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
LabelPropagation	39	886	6710	0.008319	50.860305
LabelPropagation	39	1023	2848	0.009873	21.587205
Undirect					
CutBased2+	39	963	1988	0.007493	15.068597
CutBased1+	39	907	2045	0.005388	15.500644
CutBased	39	901	2075	0.011674	15.728038
CutBased-	39	1141	1939	0.014012	14.697188
Degree					
Removing 35% of the vertices					
PageRank	45	72	2261	0.004232	17.137876
LabelPropagation	45	1061	5922	0.013542	44.887440
LabelPropagation	45	1167	2522	0.010310	19.116198
Undirect					
CutBased2+	45	1129	1583	0.010839	11.998787
CutBased1+	45	1092	1685	0.010941	12.771925
CutBased	45	1106	1705	0.011034	12.923520
CutBased-	45	1313	1512	0.015750	11.460623
Degree					
Removing 40% of the vertices					
PageRank	52	78	1614	0.003840	12.233760
LabelPropagation	52	1238	4893	0.010134	37.087850
LabelPropagation	52	1341	2110	0.008165	15.993330
Undirect					
CutBased2+	52	1343	1134	0.011396	8.595467
CutBased1+	52	1314	1275	0.011423	9.664216
CutBased	52	1305	1180	0.012464	8.944137
CutBased-	52	1474	1055	0.014240	7.996665
Degree					

Πίνακας B.2: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
Removing 45% of the vertices					
PageRank	58	78	1351	0.005685	10.240279
LabelPropagation	58	1402	4089	0.007250	30.993709
LabelPropagation	58	1443	1694	0.004187	12.840142
Undirect					
CutBased2+	58	1473	884	0.010372	6.700523
CutBased1+	58	1596	970	0.009889	7.352384
CutBased	58	1458	896	0.017620	6.791480
CutBased-	58	1601	743	0.011528	5.631774
Degree					

B.3 Πίνακες για Google

Πίνακας B.3: Αποτελέσματα της εκτέλεσης των αλγορίθμων για την αφαίρεση διαφορετικών ποσοστών κορυφών του Google. Το $f(G)$ αναφέρεται στις υπόλοιπες τιμές της αρχικής τιμής transitive closure του G . $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές, και Time είναι ο χρόνος εκτέλεσης σε δευτερόλεπτα.

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
Removing 5% of the vertices					
PageRank	789	15005	21140248	3.757233	8.734132
LabelPropagation	789	82739	62007471	20.650412	25.618500
LabelPropagation	789	107958	135138	4.403852	0.055833
Undirect					
CutBased2+	789	89000	162834	3.293006	0.067275

Πίνακας Β.3: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
CutBased1+	789	90098	164909	2.693317	0.068132
CutBased	789	89199	122381	6.667655	0.050562
CutBased-	789	93548	124952	6.453105	0.051624
Degree					
Removing 10% of the vertices					
PageRank	1577	15586	16591768	6.317747	6.854919
LabelPropagation	1577	102942	18980055	39.878125	7.841644
LabelPropagation	1577	118537	55025	7.272798	0.022734
Undirect					
CutBased2+	1577	101865	91964	5.279780	0.037995
CutBased1+	1577	101978	95225	4.710876	0.039342
CutBased	1577	104388	72738	10.401476	0.030052
CutBased-	1577	105162	74047	10.682494	0.030593
Degree					
Removing 15% of the vertices					
PageRank	2365	16837	12213076	8.643320	5.045855
LabelPropagation	2365	112082	4771028	55.438553	1.971159
LabelPropagation	2365	124618	42097	9.917473	0.017392
Undirect					
CutBased2+	2365	111274	62123	6.608010	0.025666
CutBased1+	2365	111106	68003	6.337026	0.028096
CutBased	2365	114437	50973	13.324913	0.021060
CutBased-	2365	113518	49807	13.340210	0.020578
Degree					
Removing 20% of the vertices					
PageRank	3153	17037	10173806	10.743280	4.203327
LabelPropagation	3153	124281	70904	69.029783	0.029294
LabelPropagation	3153	130760	30051	11.956968	0.012416
Undirect					

Πίνακας Β.3: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
CutBased2+	3153	120077	43012	8.108697	0.017770
CutBased1+	3153	119803	44502	7.775343	0.018386
CutBased	3153	121888	35379	15.224323	0.014617
CutBased-	3153	120385	36075	15.408039	0.014904
Degree					
Removing 25% of the vertices					
PageRank	3941	19219	8391290	12.493179	3.466877
LabelPropagation	3941	132759	35232	77.926237	0.014556
LabelPropagation	3941	136147	17421	13.862187	0.007198
Undirect					
CutBased2+	3941	127194	29682	9.118470	0.012263
CutBased1+	3941	127052	30496	9.101123	0.012599
CutBased	3941	127694	24985	17.434790	0.010323
CutBased-	3941	126881	24869	17.109108	0.010275
Degree					
Removing 30% of the vertices					
PageRank	4729	21717	5375175	13.947774	2.220763
LabelPropagation	4729	137634	23726	83.839402	0.009802
LabelPropagation	4729	140174	12849	15.233877	0.005309
Undirect					
CutBased2+	4729	133427	20161	10.224490	0.008330
CutBased1+	4729	133260	20415	10.276834	0.008434
CutBased	4729	133637	17210	18.492886	0.007110
CutBased-	4729	132941	17394	18.535501	0.007186
Degree					
Removing 35% of the vertices					
PageRank	5518	22346	3093526	15.236719	1.278096
LabelPropagation	5518	142126	14832	90.589512	0.006128

Πίνακας Β.3: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
LabelPropagation Undirect	5518	143892	9084	16.210124	0.003753
CutBased2+	5518	138783	13424	11.177761	0.005546
CutBased1+	5518	138713	13538	11.289168	0.005593
CutBased	5518	139346	11571	19.408324	0.004781
CutBased- Degree	5518	138295	11988	19.520359	0.004953
Removing 40% of the vertices					
PageRank	6306	24912	1506212	16.185635	0.622294
LabelPropagation	6306	146034	8757	93.997071	0.003618
LabelPropagation Undirect	6306	147506	5445	17.501643	0.002250
CutBased2+	6306	143221	8732	12.414031	0.003608
CutBased1+	6306	143174	8753	12.108838	0.003616
CutBased	6306	143751	7722	20.398135	0.003190
CutBased- Degree	6306	142520	8178	20.301068	0.003379
Removing 45% of the vertices					
PageRank	7094	25308	1157895	17.176632	0.478386
LabelPropagation	7094	149083	4419	95.755047	0.001826
LabelPropagation Undirect	7094	150371	2576	18.244368	0.001064
CutBased2+	7094	146937	5611	12.786246	0.002318
CutBased1+	7094	146903	5621	12.683438	0.002322
CutBased	7094	147303	5011	20.400727	0.002070
CutBased- Degree	7094	145842	5602	20.812083	0.002314

B.4 Πίνακες για LastFm Asia

Πίνακας B.4: Αποτελέσματα της εκτέλεσης των αλγορίθμων για την αφαίρεση διαφορετικών ποσοστών κορυφών του LastFm Asia. Το $f(G)$ αναφέρεται στις υπόλοιπες τιμές της αρχικής τιμής transitive closure του G . $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές, και Time είναι ο χρόνος εκτέλεσης σε δευτερόλεπτα.

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
Removing 5% of the vertices					
PageRank	382	14229	40558194	4.312355	69.786240
LabelPropagation	382	20520	41159226	4.285825	70.820403
LabelPropagation	382	24450	41454736	2.117260	71.328870
Undirect					
CutBased2+	382	7477	37633886	9.649821	64.754545
CutBased1+	382	8307	38111209	5.337571	65.575848
CutBased	382	7719	38346399	20.301585	65.980527
CutBased-	382	13230	37744110	20.325525	64.944202
Degree					
Removing 10% of the vertices					
PageRank	763	18981	27505886	6.423240	47.327856
LabelPropagation	763	30448	28362994	6.994838	48.802634
LabelPropagation	763	36394	30135564	2.391647	51.852597
Undirect					
CutBased2+	763	14664	24638569	21.533996	42.394222
CutBased1+	763	14485	24652696	10.609627	42.418530
CutBased	763	14168	24580376	41.787167	42.294093
CutBased-	763	21426	23885368	43.723337	41.098231
Degree					
Removing 15% of the vertices					
PageRank	1144	21849	14891776	8.507151	25.623455
LabelPropagation	1144	37540	7442846	9.051957	12.806493

Πίνακας Β.4: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
LabelPropagation Undirect	1144	42714	15043386	2.428938	25.884322
CutBased2+	1144	20659	14961800	29.824581	25.743941
CutBased1+	1144	20544	14544944	14.212980	25.026680
CutBased	1144	20457	14828840	59.704326	25.515164
CutBased- Degree	1144	27679	11570193	50.835615	19.908191
Removing 20% of the vertices					
PageRank	1525	23779	3244070	10.474472	5.581892
LabelPropagation	1525	43950	301306	10.605815	0.518441
LabelPropagation Undirect	1525	46966	56640	2.458086	0.097457
CutBased2+	1525	26564	6949382	35.138457	11.957417
CutBased1+	1525	27175	7280130	16.501454	12.526517
CutBased	1525	26141	6835127	65.063808	11.760825
CutBased- Degree	1525	33082	2330853	53.385132	4.010570
Removing 25% of the vertices					
PageRank	1906	25230	26420	12.582975	0.045459
LabelPropagation	1906	48332	28660	11.760312	0.049314
LabelPropagation Undirect	1906	49732	11740	2.755895	0.020200
CutBased2+	1906	32966	2062088	37.225427	3.548121
CutBased1+	1906	33399	1082406	17.568667	1.862436
CutBased	1906	32618	1206153	66.928039	2.075361
CutBased- Degree	1906	37936	31724	53.825122	0.054586
Removing 30% of the vertices					
PageRank	2288	26164	8084	14.338853	0.013910

Πίνακας Β.4: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
LabelPropagation	2288	51048	7592	12.634438	0.013063
LabelPropagation Undirect	2288	51652	5558	3.047350	0.009563
CutBased2+	2288	39267	32614	37.949268	0.056117
CutBased1+	2288	39426	29709	17.990267	0.051119
CutBased	2288	39420	26126	67.352946	0.044954
CutBased- Degree	2288	41159	16203	55.951340	0.027880
Removing 35% of the vertices					
PageRank	2669	26984	1644	15.845082	0.002829
LabelPropagation	2669	52752	3720	13.485409	0.006401
LabelPropagation Undirect	2669	52740	3904	3.224351	0.006717
CutBased2+	2669	43070	12604	38.851929	0.021687
CutBased1+	2669	42971	12465	18.429013	0.021448
CutBased	2669	43159	11576	68.920343	0.019918
CutBased- Degree	2669	43979	10100	56.087426	0.017379
Removing 40% of the vertices					
PageRank	3050	27376	860	17.092893	0.001480
LabelPropagation	3050	54106	1602	13.936888	0.002756
LabelPropagation Undirect	3050	53638	2556	3.279245	0.004398
CutBased2+	3050	46456	6738	39.781256	0.011594
CutBased1+	3050	46376	6852	18.897174	0.011790
CutBased	3050	46528	6377	68.718560	0.010973
CutBased- Degree	3050	46643	6495	56.730492	0.011176

Πίνακας B.4: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
Removing 45% of the vertices					
PageRank	3431	27757	98	18.123204	0.000169
LabelPropagation	3431	55010	618	14.094915	0.001063
LabelPropagation Undirect	3431	54526	1302	3.430284	0.002240
CutBased2+	3431	49408	4346	39.535014	0.007478
CutBased1+	3431	49285	4422	19.086116	0.007609
CutBased	3431	49421	4155	68.913704	0.007149
CutBased- Degree	3431	49323	4434	56.950795	0.007629

B.5 Πίνακες για Rome

Πίνακας B.5: Αποτελέσματα της εκτέλεσης των αλγορίθμων για την αφαίρεση διαφορετικών ποσοστών κορυφών του Rome. Το $f(G)$ αναφέρεται στις υπόλοιπες τιμές της αρχικής τιμής transitive closure του G . $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές, και Time είναι ο χρόνος εκτέλεσης σε δευτερόλεπτα.

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
Removing 5% of the vertices					
PageRank	168	606	8934349	0.447265	79.492353
LabelPropagation	168	1377	9656612	0.343429	85.918605
LabelPropagation Undirect	168	1233	9641124	0.282524	85.780803
CutBased2+	168	683	7933508	0.796373	70.587484

Πίνακας Β.5: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
CutBased1+	168	663	6660831	0.509973	59.263985
CutBased	168	687	7056805	0.933179	62.787119
CutBased-	168	697	6805640	0.875865	60.552407
Degree					
Removing 10% of the vertices					
PageRank	336	1184	6707869	0.694830	59.682500
LabelPropagation	336	2595	6670400	0.658362	59.349124
LabelPropagation	336	2417	5442977	0.295403	48.428268
Undirect					
CutBased2+	336	1375	1620703	1.499829	14.420020
CutBased1+	336	1364	3523243	0.933078	31.347653
CutBased	336	1388	1557231	1.915623	13.855285
CutBased-	336	1427	984318	1.528218	8.757857
Degree					
Removing 15% of the vertices					
PageRank	503	1716	4241166	0.938800	37.735291
LabelPropagation	503	3598	1292578	0.983474	11.500566
LabelPropagation	503	3471	401335	0.300137	3.570832
Undirect					
CutBased2+	503	2094	182079	1.697848	1.620027
CutBased1+	503	2082	566552	1.065927	5.040832
CutBased	503	2102	123245	2.174549	1.096558
CutBased-	503	2140	108816	1.714867	0.968178
Degree					
Removing 20% of the vertices					
PageRank	671	2208	846473	1.123316	7.531397
LabelPropagation	671	4595	134912	1.243201	1.200364
LabelPropagation	671	4395	53907	0.373913	0.479631
Undirect					

Πίνακας Β.5: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
CutBased2+	671	2798	55867	1.812570	0.497070
CutBased1+	671	2788	63537	1.183076	0.565313
CutBased	671	2806	42125	2.304007	0.374802
CutBased-	671	2872	37803	1.925816	0.336348
Degree					
Removing 25% of the vertices					
PageRank	839	2671	249522	1.413599	2.220094
LabelPropagation	839	5500	19636	1.520801	0.174709
LabelPropagation	839	5322	13162	0.445792	0.117107
Undirect					
CutBased2+	839	3497	26011	1.909297	0.231430
CutBased1+	839	3495	27138	1.299656	0.241457
CutBased	839	3507	20912	2.481510	0.186062
CutBased-	839	3583	19082	2.062603	0.169780
Degree					
Removing 30% of the vertices					
PageRank	1006	3117	72470	1.647253	0.644794
LabelPropagation	1006	6191	7650	1.764041	0.068065
LabelPropagation	1006	6123	5471	0.503481	0.048678
Undirect					
CutBased2+	1006	4179	12802	2.029599	0.113904
CutBased1+	1006	4168	13421	1.420950	0.119412
CutBased	1006	4200	10923	2.606542	0.097186
CutBased-	1006	4270	9724	2.202476	0.086518
Degree					
Removing 35% of the vertices					
PageRank	1174	3502	12587	1.881104	0.111991
LabelPropagation	1174	6819	3209	2.004321	0.028552

Πίνακας B.5: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
LabelPropagation Undirect	1174	6778	3018	0.562124	0.026852
CutBased2+	1174	4864	7159	2.112306	0.063696
CutBased1+	1174	4846	7239	1.516049	0.064408
CutBased	1174	4879	6603	2.748033	0.058749
CutBased- Degree	1174	4938	5828	2.329058	0.051854
Removing 40% of the vertices					
PageRank	1342	3689	10609	2.050678	0.094392
LabelPropagation	1342	7274	2311	2.174323	0.020562
LabelPropagation Undirect	1342	7254	2036	0.598737	0.018115
CutBased2+	1342	5517	4553	2.214937	0.040510
CutBased1+	1342	5520	4242	1.613170	0.037743
CutBased	1342	5550	3912	2.847791	0.034807
CutBased- Degree	1342	5608	3644	2.404029	0.032422
Removing 45% of the vertices					
PageRank	1509	3871	9669	2.224765	0.086029
LabelPropagation	1509	7699	1533	2.323397	0.013640
LabelPropagation Undirect	1509	7619	1555	0.648615	0.013835
CutBased2+	1509	6153	2763	2.276218	0.024583
CutBased1+	1509	6143	2612	1.665997	0.023240
CutBased	1509	6174	2526	2.934462	0.022475
CutBased- Degree	1509	6232	2427	2.507318	0.021594

B.6 Πίνακες για Twitter

Πίνακας B.6: Αποτελέσματα της εκτέλεσης των αλγορίθμων για την αφαίρεση διαφορετικών ποσοστών κορυφών του Twitter. Το $f(G)$ αναφέρεται στις υπόλοιπες τιμές της αρχικής τιμής transitive closure του G . $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές, και Time είναι ο χρόνος εκτέλεσης σε δευτερόλεπτα.

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
Removing 5% of the vertices					
PageRank	87	453	2205656	0.129410	74.081180
LabelPropagation	87	1573	2402170	0.185132	80.681478
LabelPropagation Undirect	87	2650	2165774	0.095832	72.741666
CutBased2+	87	646	1965855	0.244243	66.027004
CutBased1+	87	571	1989070	0.134639	66.806724
CutBased	87	585	1974576	0.343738	66.319915
CutBased-Degree	87	680	1987993	0.363680	66.770551
Removing 10% of the vertices					
PageRank	173	778	1428289	0.186136	47.971821
LabelPropagation	173	2761	1781828	0.315884	59.846105
LabelPropagation Undirect	173	3840	1314959	0.109386	44.165416
CutBased2+	173	1250	1300299	0.405505	43.673031
CutBased1+	173	1253	1305857	0.267683	43.859707
CutBased	173	1200	1289710	0.686037	43.317380
CutBased-Degree	173	1468	1093721	0.714760	36.734714
Removing 15% of the vertices					
PageRank	259	1089	685594	0.218554	23.026987
LabelPropagation	259	3565	1288161	0.427784	43.265353
LabelPropagation Undirect	259	4672	463117	0.119392	15.554671
CutBased2+	259	1951	643127	0.523462	21.600652
CutBased1+	259	1908	787217	0.342631	26.440190

Πίνακας Β.6: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
CutBased	259	1897	671725	0.936999	22.561170
CutBased-Degree	259	2210	101880	0.882638	3.421835
Removing 20% of the vertices					
PageRank	346	1317	264480	0.260585	8.883067
LabelPropagation	346	4232	655338	0.521160	22.010781
LabelPropagation	346	5294	13455	0.137318	0.451912
Undirect					
CutBased2+	346	2650	155935	0.583941	5.237376
CutBased1+	346	2632	226845	0.395865	7.619024
CutBased	346	2701	49615	1.066332	1.666415
CutBased-Degree	346	2830	13349	0.953512	0.448352
Removing 25% of the vertices					
PageRank	432	1491	56775	0.286964	1.906897
LabelPropagation	432	4782	348940	0.620512	11.719818
LabelPropagation	432	5763	3066	0.156647	0.102977
Undirect					
CutBased2+	432	3269	18222	0.624025	0.612021
CutBased1+	432	3345	18289	0.407280	0.614271
CutBased	432	3272	9271	1.125556	0.311384
CutBased-Degree	432	3371	6620	0.997348	0.222345
Removing 30% of the vertices					
PageRank	518	1648	5209	0.306182	0.174954
LabelPropagation	518	5205	139997	0.676051	4.702067
LabelPropagation	518	6110	1385	0.171790	0.046518
Undirect					
CutBased2+	518	3790	7013	0.648149	0.235545
CutBased1+	518	3883	6871	0.435055	0.230776
CutBased	518	3804	4471	1.166992	0.150167
CutBased-Degree	518	3897	3724	1.037700	0.125078
Removing 35% of the vertices					

Πίνακας Β.6: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
PageRank	605	1786	1987	0.311921	0.066737
LabelPropagation	605	5631	18135	0.742480	0.609099
LabelPropagation	605	6374	751	0.185932	0.025224
Undirect					
CutBased2+	605	4325	3641	0.678453	0.122290
CutBased1+	605	4353	3604	0.461220	0.121047
CutBased	605	4293	2769	1.215916	0.093002
CutBased-Degree	605	4443	2409	1.054939	0.080911
Removing 40% of the vertices					
PageRank	691	1885	916	0.330455	0.030766
LabelPropagation	691	5922	6918	0.793405	0.232354
LabelPropagation	691	6577	429	0.197531	0.014409
Undirect					
CutBased2+	691	4879	2226	0.709583	0.074764
CutBased1+	691	4892	2188	0.472825	0.073488
CutBased	691	4866	1805	1.202670	0.060624
CutBased-Degree	691	4977	1707	1.076159	0.057333
Removing 45% of the vertices					
PageRank	777	1937	256	0.335825	0.008598
LabelPropagation	777	6233	2171	0.836331	0.072917
LabelPropagation	777	6716	244	0.202633	0.008195
Undirect					
CutBased2+	777	5398	1463	0.724924	0.049138
CutBased1+	777	5420	1455	0.497147	0.048869
CutBased	777	5388	1261	1.222945	0.042353
CutBased-Degree	777	5514	1243	1.080956	0.041749

B.7 Πίνακες για Gnutella25

Πίνακας B.7: Αποτελέσματα της εκτέλεσης των αλγορίθμων για την αφαίρεση διαφορετικών ποσοστών κορυφών του Gnutella25. Το $f(G)$ αναφέρεται στις υπόλοιπες τιμές της αρχικής τιμής transitive closure του G . $TC(\%)$ είναι το ποσοστό των εναπομεινάντων τιμών της $f(G)$ σε σύγκριση με τις αρχικές, και Time είναι ο χρόνος εκτέλεσης σε δευτερόλεπτα.

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
Removing 5% of the vertices					
PageRank	258	696	21736087	1.321834	81.905669
CutBased2+	258	1323	21301322	4.187961	80.267392
LabelPropagation	258	1478	23162605	3.711498	87.281057
LabelPropagation	258	3887	22097947	0.794654	83.269225
Undirect					
CutBased1+	258	1261	21474591	1.808471	80.920302
CutBased	258	1298	21375170	2.823410	80.545665
CutBased-Degree	257	1401	21311528	2.958426	80.305850
Removing 10% of the vertices					
PageRank	516	1260	17669504	1.767376	66.582018
CutBased2+	516	2683	16353183	8.387308	61.621873
LabelPropagation	516	3051	20014773	8.090187	75.419433
LabelPropagation	516	6579	17615346	0.941039	66.377941
Undirect					
CutBased1+	516	2594	16636370	3.820748	62.688975
CutBased	516	2621	16455704	6.288434	62.008191
CutBased-Degree	515	2825	16062908	6.333224	60.528062
Removing 15% of the vertices					
PageRank	772	1730	13988266	2.142909	52.710420
CutBased2+	773	4043	11444881	11.849523	43.126467
LabelPropagation	772	4798	17001607	12.949878	64.065256
LabelPropagation	772	8765	13002119	1.044635	48.994433
Undirect					
CutBased1+	773	4007	11782285	5.433674	44.397868

Πίνακας Β.7: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
CutBased	773	3960	11382242	9.106189	42.890431
CutBased-Degree	772	4290	9966911	9.214161	37.557197
Removing 20% of the vertices					
PageRank	1031	2134	10569081	2.404104	39.826287
CutBased2+	1031	5477	5526981	14.227439	20.826705
LabelPropagation	1031	7202	14024736	16.151630	52.847846
LabelPropagation	1031	10620	7607859	1.124515	28.667845
Undirect					
CutBased1+	1031	5454	5645932	6.692550	21.274935
CutBased	1031	5441	4116140	10.970045	15.510391
CutBased-Degree	1031	5822	803757	10.347859	3.028708
Removing 25% of the vertices					
PageRank	1288	2480	7501815	2.662447	28.268251
CutBased2+	1288	7446	1039881	14.553900	3.918467
LabelPropagation	1288	9227	10936746	17.933691	41.211718
LabelPropagation	1288	12184	2042936	1.196778	7.698168
Undirect					
CutBased1+	1288	7425	1336746	7.009550	5.037111
CutBased	1288	7315	450573	11.406100	1.697844
CutBased-Degree	1288	7419	107983	10.855968	0.406900
Removing 30% of the vertices					
PageRank	1546	2778	4896487	2.834401	18.450885
CutBased2+	1546	8492	86721	14.955670	0.326781
LabelPropagation	1546	10873	8078789	19.306000	30.442398
LabelPropagation	1546	13488	30707	1.334735	0.115710
Undirect					
CutBased1+	1546	8560	76625	7.307980	0.288737
CutBased	1546	8528	64174	11.712179	0.241820
CutBased-Degree	1546	8830	37276	11.285799	0.140463
Removing 35% of the vertices					

Πίνακας Β.7: (Continued)

Algorithm	Nodes deleted	Edges deleted	$f(G)$	Time	TC(%)
PageRank	1804	3020	2730481	3.019245	10.288967
CutBased2+	1804	9895	36910	15.205287	0.139084
LabelPropagation	1804	12298	5494914	20.712605	20.705871
LabelPropagation	1804	14571	9624	1.475760	0.036265
Undirect					
CutBased1+	1804	9973	35151	7.449761	0.132456
CutBased	1804	9926	27953	12.161041	0.105332
CutBased-Degree	1804	10220	19991	11.353132	0.075330
Removing 40% of the vertices					
PageRank	2061	3270	924987	3.181332	3.485525
CutBased2+	2061	11258	19027	15.293809	0.071697
LabelPropagation	2061	13527	2522734	21.832447	9.506137
LabelPropagation	2061	15461	4354	1.594587	0.016407
Undirect					
CutBased1+	2061	11317	18960	7.650716	0.071445
CutBased	2061	11304	15109	12.241973	0.056934
CutBased-Degree	2061	11576	11621	11.461360	0.043790
Removing 45% of the vertices					
PageRank	2319	3458	255211	3.331432	0.961683
CutBased2+	2319	12613	10839	15.493521	0.040843
LabelPropagation	2319	14557	134550	22.847705	0.507010
LabelPropagation	2319	16231	2075	1.717906	0.007819
Undirect					
CutBased1+	2319	12651	10889	7.714263	0.041032
CutBased	2319	12657	9373	12.411285	0.035319
CutBased-Degree	2319	12952	7385	11.622619	0.027828

ΒΙΟΓΡΑΦΙΚΟ ΣΗΜΕΙΩΜΑ

Ο Γραμμένος Φώτιος γεννήθηκε τον Ιούλιο στα Γρεβενά το 1998. Τον Οκτώβριο του 2016 γράφτηκε στο προπτυχιακό πρόγραμμα του Τμήματος Επιστήμης Υπολογιστών του Πανεπιστημίου Κρήτης και αποφοίτησε τον Ιούλιο του 2021. Τον Φεβρουάριο του 2022 εγγράφηκε στο μεταπτυχιακό πρόγραμμα του Τμήματος Επιστήμης και Μηχανικής Υπολογιστών του Πανεπιστημίου Ιωαννίνων. Τα ερευνητικά του ενδιαφέροντα εστιάζονται στη θεωρία γραφημάτων, στους αλγόριθμους και στις δομές δεδομένων. Επιπλέον, από τον Μάρτιο του 2022 έως σήμερα, εργάζεται ως web developer στην εταιρεία EpsilonNet Singularlogic.