NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES

DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

PROGRAM OF POSTGRADUATE STUDIES

PhD THESIS

Selfish Behavior and Compact Representation in Routing and Information Networks

Aikaterini P. Papakonstantinopoulou

ATHENS

JANUARY 2015

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Ιδιοτελής Συμπεριφορά και Συμπαγής Αναπαράσταση σε Δίκτυα Δρομολόγησης και Πληροφόρησης

Αικατερίνη Π. Παπακωνσταντινοπούλου

ΑΘΗΝΑ

ΙΑΝΟΥΑΡΙΟΣ 2015

PhD THESIS

Selfish Behavior and Compact Representation in Routing and Information Networks

Aikaterini P. Papakonstantinopoulou

SUPERVISOR: Elias Koutsoupias, Professor UoA

THREE-MEMBER ADVISORY COMMITTEE:

Elias Koutsoupias, Professor UoA
Ioannis Emiris, Professor UoA
Stavros Kolliopoulos, Associate Professor UoA

SEVEN-MEMBER EXAMINATION COMMITTEE

Elias Koutsoupias
Professor UoA

Ioannis Emiris
Professor UoA

Stavros Kolliopoulos
Associate Professor UoA

Vassilis Zissimopoulos
Professor UoA

Aggelos Kiayias
Associate Professor UoA

Vangelis Markakis
Assistant Professor AUEB

Dimitris Fotakis
Assistant Professor NTUA

Examination Date: 21/1/2015

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Ιδιοτελής Συμπεριφορά και Συμπαγής Αναπαράσταση
σε Δίκτυα Δρομολόγησης και Πληροφόρησης

Αικατερίνη Π. Παπακωνσταντινοπούλου

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Ηλίας Κουτσουπιάς, Καθηγητής Ε.Κ.Π.Α.

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:
Ηλίας Κουτσουπιάς, Καθηγητής Ε.Κ.Π.Α.
Ιωάννης Εμίρης, Καθηγητής Ε.Κ.Π.Α.
Σταύρος Κολλιόπουλος, Αναπληρωτής Καθηγητής Ε.Κ.Π.Α.

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Ηλίας Κουτσουπιάς
Καθηγητής Ε.Κ.Π.Α.

Ιωάννης Εμίρης
Καθηγητής Ε.Κ.Π.Α.

Σταύρος Κολλιόπουλος
Αναπληρωτής Καθηγητής Ε.Κ.Π.Α.

Βασίλειος Ζησιμόπουλος
Καθηγητής Ε.Κ.Π.Α.

Άγγελος Κιαγιάς
Αναπληρωτής Καθηγητής Ε.Κ.Π.Α.

Ευάγγελος Μαρκάκης
Επίκουρος Καθηγητής Α.Σ.Ο.Ε.Ε.

Δημήτριος Φωτάκης
Επίκουρος Καθηγητής Ε.Μ.Π.

Ημερομηνία εξέτασης: 21/1/2015

Abstract

During the last fifteen years, Computer Science deals systematically with networks, not only from a systems' but also from a theoretical point of view. The reason is that there are networks with a huge number of users, which play a crucial role in our everyday life, hence the issues that arise in these networks are usually challenging and have to be faced accurately and efficiently. In this thesis we study networks, focusing on two main such classes of problems, namely problems that arise due to the selfish nature of the network users, and problems concerning the representation of such networks in systems' level.

The contribution of this thesis lies along three directions. As far as the selfish behavior of the network users' is concerned, we first study congestion games, which is considered the most appropriate approach for the theoretical study of congestion networks. We introduce and study the class of congestion games with time-dependent strategies. In a more practical level, we study the largest information network in our days, the worldwide web, analyzing its game-theoretic aspects. Both these objects are frameworks for studying classes of problems. We then move to the important technical issue of representing such networks so that they can fit in the computer's main memory, where we introduce a network compression algorithm that outperforms the state-of-the-art method. Our compression method is appropriate for networks created by human activity, with routing networks, social networks and the worldwide web being the most popular examples.

In order to study the users' selfish behavior, we use the Game Theory framework. Game Theory studies economic games, that is situations in which two ore more selfish entities interact. Based on this framework, we introduce congestion games in which players decide the time at which they will participate, and consider networks consisted of parallel links, and two models for the latency functions on the links. We focus on games with symmetric players; we study their structural properties and compute their Nash equilibria as well as their prices of anarchy and stability to assess the quality of the latter. We prove that some of these games, although related to congestion games, are not congestion games themselves. Moreover we prove that they possess a unique Nash equilibrium, in which players are more aggressive than in the optimal setting, entering the game with larger probabilities at early time steps. However, selfishness does not cost the players much: the price of anarchy is only 1.06. We then study the

worldwide web, considering it as the outcome of a game among the web page authors who establish hyperlinks, aiming at the maximization of their page's reputation. We focus on advertising links, whose establishment incurs some cost for the receiving page, and constitute a common strategy in Search Engine Optimization. We prove that the computation of the players' best response strategy is NP-hard and compute an approximate best response.

Regarding the compact network representation, we observed that various network classes of interest, for example routing, information and social networks, which are traditionally modeled as graphs, have a common property: they exhibit high concentration of edges around the main diagonal of the graph's adjacency matrix, probably after some reordering of the nodes of the graph. Hence we isolate the dense part of the graph and design a hybrid compression algorithm that treats the dense part differently than the rest of the graph. We provide analysis and experimental evaluation of our method on a real dataset, which show that it outperforms the currently best method by achieving a better compression ratio and retrieval time of the graph's elements.

SUBJECT AREA: Algorithmic Game Theory

KEYWORDS: Nash equilibria computation, price of anarchy, selfish routing, worldwide web, compact graph representation

Περίληψη

Κατά τη διάρκεια των τελευταίων δεκαπέντε ετών, η Επιστήμη της Πληροφορικής ασχολείται συστηματικά με τα δίκτυα, όχι μόνο από την οπτική γωνία των συστημάτων αλλά και την αντίστοιχη θεωρητική. Αυτό οφείλεται στο ότι υπάρχουν δίκτυα με τεράστιο αριθμό χρηστών, που διαδραματίζουν κρίσιμο ρόλο στην καθημερινή μας ζωή, και συνεπώς τα προβλήματα που ανακύπτουν σε τέτοια δίκτυα είναι συνήθως απαιτητικά και πρέπει να αντιμετωπίζονται με ακρίβεια και αποτελεσματικότητα. Στην παρούσα διατριβή μελετάμε δίκτυα, επικεντρωνόμενοι σε δύο κύριες τέτοιες κατηγορίες προβλημάτων, δηλαδή προβλήματα που ανακύπτουν εξαιτίας της ιδιοτελούς φύσης των χρηστών τους, και προβλήματα που αφορούν στην αναπαράσταση τέτοιων δικτύων στο επίπεδο των συστημάτων.

Η συνεισφορά αυτής της διατριβής εκτείνεται προς τρεις κατευθύνσεις. Όσον αφορά την ιδιοτελή συμπεριφορά των χρηστών δικτύων, μελετάμε αρχικά παίγνια συμφόρησης, προσέγγιση η οποία θεωρείται η πλέον αρμόζουσα για τη θεωρητική μελέτη δικτύων συμφόρησης. Εισάγουμε και μελετάμε την κλάση των παιγνίων συμφόρησης με στρατηγικές εξαρτώμενες από το χρόνο. Σε ένα πιο πρακτικό επίπεδο, μελετάμε το μεγαλύτερο δίκτυο πληροφόρησης των ημερών μας, τον παγκόσμιο ιστό, αναλύοντας τις παιγνιο-θεωρητικές πτυχές του. Και τα δύο αυτά αντικείμενα είναι πλαίσια για τη μελέτη κατηγοριών προβλημάτων. Έπειτα προχωρούμε με το σημαντικό τεχνικό θέμα της αναπαράστασης τέτοιων δικτύων ώστε να καταστεί δυνατή η απεικόνισή τους στην κύρια μνήμη, όπου παρουσιάζουμε έναν αλγόριθμο συμπίεσης δικτύων που ξεπερνά τις επιδόσεις της state-of-the-art μεθόδου. Η μέθοδος συμπίεσής μας είναι κατάλληλη για δίκτυα που έχουν δημιουργηθεί από ανθρώπινη δραστηριότητα, με τα δίκτυα δρομολόγησης, τα κοινωνικά δίκτυα και τον παγκόσμιο ιστό να είναι τα πιο δημοφιλή παραδείγματα.

Για να μελετήσουμε την ιδιοτελή συμπεριφορά των χρηστών, χρησιμοποιούμε το πλαίσιο της Θεωρίας Παιγνίων. Η Θεωρία Παιγνίων μελετά οικονομικά παίγνια, δηλαδή καταστάσεις στις οποίες δύο ή περισσότερες ιδιοτελείς οντότητες αλληλεπιδρούν. Βασιζόμενοι σε αυτό το πλαίσιο, εισάγουμε τα παίγνια συμφόρησης στα οποία οι παίκτες αποφασίζουν τη στιγμή κατά την οποία θα συμμετάσχουν, και θεωρούμε δίκτυα αποτελούμενα από παράλληλα μονοπάτια, και δύο μοντέλα για τις συναρτήσεις καθυστέρησης στα μονοπάτια αυτά. Επικεντρωνόμενοι σε παίγνια με συμμερικούς παίκτες, μελετούμε τις δομικές τους ιδιότητες και υπολογίζουμε τα σημεία ισορροπίας Nash, καθώς και τα τιμήματα αναρχίας και σταθερότητάς τους, ώστε να εκτιμήσουμε την ποιότητα των σημείων ισορροπίας. Αποδεικνύουμε πως μερικά από αυτά τα παίγνια, παρόλο που

μοιάζουν πολύ με παίγνια συμφόρησης, στην πραγματικότητα δεν είναι. Επιπρόσθετα αποδεικνύουμε πως έχουν μοναδικό σημείο ισορροπίας Nash, στο οποίο οι παίκτες είναι πιο επιθετικοί από ό,τι στη βέλτιστη δυνατή διευθέτηση του συστήματος, μπαίνοντας στο παίγνιο με μεγαλύτερες πιθανότητες στα πρώτα στάδια. Ωστόσο, η ιδιοτέλεια δεν κοστίζει πολύ στους παίκτες: το τίμημα της αναρχίας είναι μόλις 1.06. Στη συνέχεια μελετάμε τον παγκόσμιο ιστό, θεωρώντας τον ως το αποτέλεσμα ενός παιγνίου μεταξύ των συγγραφέων ιστοσελίδων που τοποθετούν υπερσυνδέσμους στη σελίδα τους, στοχεύοντας στη μεγιστοποίηση της φήμης της. Εστιάζουμε στη μελέτη της τοποθέτησης διαφημιστικών συνδέσμων, που συνεπάγονται κάποιο κόστος για τη σελίδα που τους δέχεται, η οποία αποτελεί συνήθη στρατηγική στη βελτιστοποίηση ιστοσελίδων ώστε να αυξάνει η ορατήτητά τους από τις μηχανές αναζήτησης. Αποδεικνύουμε πως ο υπολογισμός της στρατηγικής βέλτιστης απόκρισης των παιχτών είναι NP-hard και υπολογίζουμε μια προσεγγιστική βέλτιστη απόκριση.

Όσον αφορά στη συμπαγή αναπαράσταση δικτύων, παρατηρούμε πως οι διάφορες κατηγορίες δικτύων που μας ενδιαφέρουν, όπως για παράδειγμα δίκτυα δρομολόγησης, πληροφόρησης και τα κοινωνικά δίκτυα, τα οποία παραδοσιακά αναπαριστώνται ως γράφοι, έχουν μία κοινή ιδιότητα: επιδεικνύουν υψηλή συγκέντρωση ακμών γύρω από την κεντρική διαγώνιο του πίνακα γειτνίασης του γράφου, πιθανώς μετά από κάποια αναδιάταξη των κόμβων του γράφου. Ως εκ τούτου απομονώνουμε το πυκνό κομμάτι του γράφου και σχεδιάσουμε έναν υβριδικό αλγόριθμο συμπίεσης που μεταχειρίζεται το πυκνό κομμάτι διαφορετικά από ό,τι τον υπόλοιπο γράφο. Παρέχουμε ανάλυση και πειραματική αξιολόγηση της μεθόδου σε ένα αληθινό σύνολο δεδομένων, που δείχνουν πως ξεπερνά σε επιδόσεις την τρέχουσα καλύτερη μέθοδο επιτυγχάνοντας καλύτερο λόγο συμπίεσης και χρόνο προσπέλασης των στοιχείων του γράφου.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Αλγοριθμική Θεωρία Παιγνίων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Υπολογισμός σημείων ισορροπίας Nash, τίμημα της αναρχίας, ιδιοτελής δρομολόγηση, παγκόσμιος ιστός, συμπαγής αναπαράσταση γράφων

To my parents, Sofia and Panagiotis,
my brother Charalampos,
and to everyone who pursues their goals
despite facing tough -or often invincible- adversaries.

Acknowledgements

As this step of my education gets completed, I would like to thank all the people who had a positive influence on my way of living, thinking and working throughout these years.

First of all, I would like to thank Elias Koutsoupias for introducing me to Theory and making my doctoral studies an invaluable experience. Through the discussions with Elias I learnt to think simply and appreciate clear explanations of complex phenomena. His abilities in identifying important practical problems and analyzing their fundamental principles made me decide to pursue a PhD in Theoretical Computer Science. I am glad because I am still interested in this research area as much as I was at the beginning. Elias, thank you for everything I learnt because of you!

I owe a lot to my friends and collaborators Panagiotis Liakos and Michael Sioutis. We started as office mates working on entirely different aspects of Computer Science, but appreciating important practical problems with interesting theoretical aspects, and ended up conducting unsupervised research in the field of graph compression. This was an extremely interesting, fun, didactic and rewarding process, as I practiced in collaborating with people of different backgrounds and gaining expertise on an completely new subject for me. It is a pleasure working with them and I am looking forward our next research goal. Moreover, Panagiotis and Mike were always there for discussing my work, proofreading my manuscripts and attending my practice talks. Thank you guys!

I am also grateful to the members of my examination committee Ioannis Emiris, Stavros Kolliopoulos, Vassilis Zissimopoulos, Aggelos Kiayias, Vangelis Markakis and Dimitris Fotakis for the fruitful discussions I had with them on topics of my thesis. Special acknowledgements go to Vassilis Zissimopoulos for his warm attitude towards me and his real interest on my progress throughout these years, and to Dimitris Fotakis for motivating me with his enthousiasm with research every time we discuss.

I feel lucky for the really talented and modest persons I met during these years. During the Algorithmic Game Theory semester at Hebrew University, Jerusalem, I had the priviledge to spend time with Professors Anna Karlin, Amos Fiat, and Elias, and I will never forget the inspiring environment they formed wherever they were, even when they were discussing trivial issues. The memories of the weekends in Jaffa and the view around Andromeda's rock come naturally to me then I feel calm and fulfilled while studying. Christos Papadimitriou, working with the enthusiasm of a 22-year old

student a few of times we collaborated, was another positive influence for me. I became motivated in pursuing a PhD while studying part of the work of Jon Kleinberg during my early postgraduate studies. I admired his inspiring work on many diverse areas.

I also appreciate deeply the influence of Niki Kontaxaki, my ballet teacher, who taught us to set goals and work hard to achieve them, and Elissavet Dionyssopoulou, my engraving teacher, because next to her I learnt to illustrate my thoughts, see from different aspects, and develop the sense of symmetry.

I always enjoyed short discussions with friends that started with a simple "how are you doing?", from friends being interested enough about the answer and willing to process it, since they were giving me the delight to share my thoughts and motivating me further: Orestis, thanks for posing nice questions to me, Christos, thanks for the afternoon research sessions, Tali, thanks for caring! During the last year of my studies I was sharing an office with the programming languages research group led by Professor Yiannis Smaragdakis. I thank Yiannis for considering me a 'honorary member' of their group and treating me like I was more than it, asking me to describe him the problems I was working on, in some cases after many successive hours of teaching and meetings. I am also grateful to my non-academic friends, especially Foteini, Foteini and Vasso, for keeping in touch when I had no time to even think about it.

My research was partially funded by the Greek State Scholarships Foundation, and the research programs AEOLUS and THALES. These programs gave me the opportunity to interact with various researchers and broaden my understanding on my field of research.

Finally, I want to thank my family. My parents and brother supported me in all possible ways during my studies and were encouraging me in whatever I pursued. The sense of freedom I had at home, allowed me to make decisions regarding my studies for which I am truly happy now. However, it wouldn't be the same without the support of Alexandros. His love, mentality and activity make me happy and inspire me. I wish he continues rocking, in Theory and in practice.

Συνοπτική Παρουσίαση της Διδακτορικής Διατριβής

Εισαγωγή

Κατά τη διάρκεια των τελευταίων δεκαπέντε ετών, η εκρηκτική ανάπτυξη του Διαδικτύου και η χρήση του στην καθημερινή ζωή έχει οδηγήσει σε διάφορους τύπους δικτύων που δημιουργούνται από την ανθρώπινη δραστηριότητα, χωρίς κανένα κεντρικό σχεδιασμό και έλεγχο. Το πιο γνωστό παράδειγμα είναι ο παγκόσμιος ιστός (WWW), μια υπηρεσία χτισμένη πάνω από το διαδίκτυο, που επιτρέπει στους χρήστες της να έχουν πρόσβαση σε πληροφορίες από υπολογιστές που μπορούν να βρίσκονται οπουδήποτε στον κόσμο, χρησιμοποιώντας μια εξαιρετικά απλή διεπαφή. Φυσικά, ο παγκόσμιος ιστός ξεκίνησε πολλά χρόνια νωρίτερα, γύρω στα μέσα της δεκαετίας του '80, αλλά έγινε αντικείμενο μελέτης όταν άρχισε να χρησιμοποιείται από μεγάλο αριθμό χρηστών, προσελκύοντας έτσι έντονη δραστηριότητα σχετικά με αναζήτηση πληροφορίας αλλά και οικονομική δραστηριότητα. Πιο πρόσφατα, ένα σύνολο από νέα δίκτυα, διαθέσιμα μέσω του διαδικτύου, έχουν αναπτυχθεί, που ενσωματώνουν πληροφορία από το κοινωνικό περιβάλλον των χρηστών, και αναφέρονται ως κοινωνικά δίκτυα. Η ανάπτυξη του Διαδικτύου έχει δώσει κίνητρα και για την μελέτη της δρομολόγησης σε δίκτυα, καθώς συνήθως υπάρχουν διάφοροι εναλλακτικοί τρόποι για την επικοινωνία μεταξύ δύο οντοτήτων ενός δικτύου.

Η δημοτικότητα αυτών των δικτύων έχει σαφώς επηρεάσει τα κίνητρα των χρηστών τους. Για παράδειγμα, παρόλο που η κύρια σκέψη του συγγραφέα μιας ιστοσελίδας όταν αποφάσιζε για τους υπερσυνδέσμους που θα συνέδεαν τη σελίδα του με το WWW ήταν να δημιουργήσει μια σελίδα με *υψηλή ποιότητα περιεχομένου*, λίγα χρόνια αργότερα, η *δημοτικότητα* της σελίδας του, η οποία μπορεί εύκολα να μετατραπεί σε οικονομικό όφελος, έγινε το κύριο μέλημά του. Ωστόσο, η προσέλκυση των χρηστών σε κάποιο πόρο του δικτύου δεν είναι επιθυμητή σε όλους τους τύπους δικτύων. Για παράδειγμα, στη δρομολόγηση σε δίκτυα είναι συνήθως προς το συμφέρον του χρήστη το να *αποφύγει* τα κανάλια επικοινωνίας που παρουσιάζουν συμφόρηση. Προφανώς, δεδομένου ότι θέλουμε να μπορούμε να κατανοούμε τους μηχανισμούς δημιουργίας τέτοιων δικτύων, η ανάλυση δικτύων έχει γίνει ένα πολύ ενδιαφέρον αντικείμενο μελέτης.

Εκτός από τις συνέπειες της δημοτικότητας των δικτύων που αντιλαμβανόμαστε από τη σκοπιά του χρήστη, υπάρχουν και συνέπειες που βλέπουμε από την πλευρά του συστή-

ματος. Ίσως το πρωταρχικό ζήτημα αυτής της κατηγορίας είναι ότι οι αναπαραστάσεις του δικτύου δεν μπορούν να χωρέσουν στην κύρια μνήμη ενός υπολογιστή, οδηγώντας σε δύο προβλήματα: τον περιορισμό του μεγέθους του δικτύου που μπορούμε να διαχειριστούμε, και τη μειωμένη απόδοση κρίσιμων εφαρμογών που τρέχουν πάνω από τέτοιες δικτυακές υποδομές, που επιφέρει η αποθήκευση της αναπαράστασης του δικτύου στη δευτερεύουσα μνήμη. Κανένα από αυτά δεν είναι αποδεκτό, οπότε χρειαζόμαστε συμπιεσμένες αναπαραστάσεις. Ωστόσο, η μεγάλη συμπίεση μπορεί να καταστήσει την ανάκτηση των στοιχείων του δικτύου πολύπλοκη, και άρα χρονοβόρα διαδικασία. Έτσι, χρειαζόμαστε συμπιεσμένες αναπαραστάσεις του δικτύου που όμως επιτρέπουν γρήγορη πρόσβαση στα στοιχεία του δικτύου.

Η διατριβή αυτή διεξήχθη σύμφωνα με τις δύο βασικές κατευθύνσεις που σκιαγραφήθηκαν παραπάνω. Η πρώτη είναι η μελέτη της *εγωιστικής συμπεριφοράς* σε διάφορα δίκτυα που παρουσιάζουν ιδιαίτερο ενδιαφέρον στις μέρες μας, ιδίως στα δίκτυα δρομολόγησης και στα δίκτυα πληροφόρησης. Και στους δύο αυτούς τύπους κάθε χρήστης προσπαθεί να μεγιστοποιήσει το δικό όφελος από το δίκτυο, ωστόσο τα διαφορετικά κίνητρα καθιστούν απαραίτητη τη μελέτη κάθε τύπου δικτύου χωριστά. Για δίκτυα δρομολόγησης, τα οποία μελετώνται θεωρητικά χρησιμοποιώντας τα παίγνια συμφόρησης, εισάγουμε και να μελετάμε την κλάση των παιγνίων συμφόρησης με τις στρατηγικές εξαρτώμενες από το χρόνο. Σε ένα πιο πρακτικό επίπεδο, μοντελοποιούμε και μελετάμε τον παγκόσμιο δίκτυο, που είναι το πιο σημαντικό δίκτυο πληροφοριών σήμερα, από ένα παιγνιοθεωρητική άποψη.

Η δεύτερη κατεύθυνση έχει να κάνει με ένα σημαντικό τεχνικό ζήτημα, την αναπαράσταση δικτύων και πιο συγκεκριμένα με τη *συμπαγή αλλά και αποτελεσματική αναπαράσταση*, που μπορεί να ενισχύσει την απόδοση των κρίσιμων εφαρμογών που εκτελούνται σε τέτοια δίκτυα. Παρατηρούμε τις κοινές ιδιότητες των δικτύων που μας ενδιαφέρουν και τις εκμεταλλευόμαστε για να σχεδιάσουμε μια αποδοτική μέθοδο συμπίεσης.

Συνεισφορά της διατριβής

Η συνεισφορά της διατριβής αυτής συνοψίζεται στα εξής:

- εισάγουμε τη διάσταση του χρόνου σε παίγνια συμφόρησης, προτείνοντας μια κλάση παιγνίων στα οποία οι παίκτες αποφασίζουν όχι μόνο το σύνολο των πόρων που πρόκειται να χρησιμοποιήσουν, αλλά και τη χρονική στιγμή που θα μπουν στο παιχνίδι,

- μελετάμε τις παιγνιοθεωρητικές όψεις της δημιουργίας υπερσυνδέσμων στον παγκόσμιο δίκτυο,

- προτείνουμε μια μέθοδο συμπίεσης γράφων για γράφους δικτύων που δημιουργούνται από την ανθρώπινη δραστηριότητα, η οποία υπερτερεί της τρέχουσας state-of-the-art μεθόδου.

Στα τρία εδάφια που ακολουθούν παρουσιάζουμε τις παραπάνω περιοχές, και συνοψίζουμε με τα συμπεράσματα και ενδιαφέροντα ανοιχτά προβλήματα που εντοπίσαμε.

Επιλέγοντας το χρόνο της δρομολόγησης

Τα τελευταία δώδεκα χρόνια, οι έννοιες του τιμήματος της αναρχίας (PoA) και της σταθερότητας (PoS) έχουν εφαρμοστεί με επιτυχία σε πολλές κατηγορίες παιγνίων, κυρίως στα παίγνια συμφόρησης και άλλα παρόμοια [91, 115, 103]. Στα παίγνια συμφόρησης, οι παίκτες ανταγωνίζονται μεταξύ τους για ένα σύνολο πόρων, όπως εγκαταστάσεις ή συνδέσεις, το κόστος του κάθε παίκτη εξαρτάται από το πλήθος των παικτών που χρησιμοποιούν τους ίδιους πόρους, και γίνεται η παραδοχή είναι ότι κάθε πόρος μπορεί να μοιραστεί μεταξύ των παικτών, αλλά με κάποιο κόστος. Μια άλλη ενδιαφέρουσα κλάση παιγνίων είναι τα παίγνια ανταγωνισμού (contention) [66], στα οποία οι παίκτες και πάλι ανταγωνίζονται για τους πόρους, αλλά οι πόροι δεν είναι δυνατό να διαμοιραστούν. Αν περισσότεροι από ένας παίκτες προσπαθήσουν να διαμοιραστούν έναν πόρο την ίδια στιγμή, ο πόρος καθίσταται μη διαθέσιμος και οι παίκτες πρέπει να ξαναδοκιμάσουν αργότερα. Υπάρχουν, ωστόσο, ενδιαφέροντα παιχνίδια που βρίσκονται μεταξύ των δύο ακραίων περιπτώσεων των παιγνίων συμφόρησης και των παιγνίων ανταγωνισμού. Για παράδειγμα, το παιχνίδι που παίζουν οι χρήστες για την αντιμετώπιση της συμφόρησης σε ένα δίκτυο φαίνεται να βρίσκεται ανάμεσα στις δύο αυτές κλάσεις – η TCP πολιτική ελέγχου συμφόρησης είναι μια στρατηγική αυτού του παιχνιδιού. Ο χρόνος είναι μέρος της στρατηγικής των παικτών (όπως στα παίγνια ανταγωνισμού) και η καθυστέρηση ενός μονοπατιού εξαρτάται από το πόσοι παίκτες χρησιμοποιούν τις ακμές του (όπως στα παίγνια συμφόρησης).

Σε αυτή τη διατριβή, προσπαθούμε να εντοπίσουμε τα βασικά χαρακτηριστικά αυτών των παιχνιδιών, να τα μοντελοποιήσουμε, και να μελετήσουμε τις ιδιότητές τους, τις Nash ισορροπίες τους, και τα τιμήματα αναρχίας και σταθερότητάς τους. Τα παίγνια που θεωρούμε είναι ουσιαστικά παίγνια συμφόρησης με την προσθήκη της διάστασης του χρόνου. Η διαφορά με τα κλασσικά παίγνια συμφόρησης είναι ότι οι παίκτες τώρα δεν επιλέγουν απλά ποια διαδρομή να χρησιμοποιήσουν, αλλά αποφασίζουν και πότε θα ξεκινήσει η μετάδοση.

Θεωρούμε μια ακμή e ενός παιγνίου συμφόρησης με συνάρτηση καθυστέρησης ℓ_e. Κάθε παίκτης που χρησιμοποιεί την ακμή e υφίσταται καθυστέρηση $\ell_e(k)$, όπου k είναι το πλήθος των παικτών που χρησιμοποιούν την ακμή. Δεδομένου ότι οι παίκτες μπορούν να αποφασίσουν πότε θα ξεκινήσουν, η καθυστέρηση πρέπει να επαναοριστεί. Ορίζουμε

και μελετάμε δύο μοντέλα καθυστέρησης για τους συνδέσμους:

Το μοντέλο boat: στο οποίο μόνο η ομάδα των παικτών που ξεκινούν ταυτόχρονα επηρεάζει την καθυστέρηση της ομάδας: φανταστείτε ότι μια βάρκα ξεκινάει από την αρχή μιας ακμής σε κάθε χρονικό βήμα, όλοι οι παίκτες που αποφασίζουν να ξεκινήσουν τη χρονική στιγμή t μπαίνουν στη βάρκα που θα τους μεταφέρει στην προορισμό τους, και η ταχύτητα της βάρκας εξαρτάται μόνο από το πλήθος των παικτών που βρίσκονται μέσα και είναι ανεξάρτητη από τους παίκτες στις άλλες βάρκες. βάρκες.

Το μοντέλο conveyor belt: στο οποίο η καθυστέρηση ενός παίκτη εξαρτάται από το πλήθος των παικτών που χρησιμοποιούν την ίδια ακμή συγχρόνως, ανεξάρτητα από το εάν ξεκίνησαν νωρίτερα ή αργότερα. Πιο συγκεκριμένα, ο σύνδεσμος είναι σαν ένας κυλιόμενος διάδρομος από την πηγή μέχρι τον προορισμό και η ταχύτητα του διαδρόμου κάθε στιγμή εξαρτάται από το πλήθος των ανθρώπων σε αυτόν.

Στην εργασία αυτή, θεωρούμε απλά δίκτυα, δηλαδή ένα σύνολο από παράλληλες συνδέσεις με γραμμικές καθυστερήσεις, μη-προσαρμοστικές (non-adaptive) στρατηγικές, στις οποίες οι παίκτες αποφασίζουν τη στρατηγική εκ των προτέρων, και συμμετρικές στρατηγικές.

Μελετάμε αρχικά τις δομικές ιδιότητες των παιγνίων boat και conveyor belt. Δείχνουμε ότι τα παίγνια boat είναι παίγνια συμφόρησης. Αντίθετα, δίνουμε παραδείγματα που δείχνουν ότι τα παίγνια conveyor belt δεν είναι στη γενική περίπτωση παίγνια συμφόρησης, με εξαίρεση την περίπτωση των δύο παικτών. Στην πραγματικότητα, ακόμα και απλά παιχνίδια με 3 παίκτες ενδέχεται να μην έχουν απλές ισορροπίες Nash.

Στη συνέχεια χαρακτηρίζουμε τις συμμετρικές ισορροπίες Nash του μοντέλου boat για παράλληλες συνδέσεις με affine συναρτήσεις καθυστέρησης, δηλαδή, $\ell_e(k) = a_e\, k + b_e$, και οποιοδήποτε πλήθος παικτών. Δείχνουμε ότι υπάρχει μια μοναδική συμμετρική μικτή ισορροπία Nash γι αυτά τα παιχνίδια. Στο σημείο ισορροπίας Nash η πιθανότητα ένας παίκτης να ξεκινά τη χρονική στιγμή t πέφτει γραμμικά με το t. Επίσης, υπολογίζουμε τη βέλτιστη συμμετρική λύση, η οποία προκύπτει ότι έχει παρόμοια μορφή με την ισορροπία Nash με τη διαφορά ότι τώρα οι παίχτες είναι λιγότερο επιθετικοί, βάζοντας μικρότερη πιθανότητα στα πρώτα στάδια του παιχνιδιού. Από το χαρακτηρισμό της ισορροπιών Nash και της βέλτιστης στρατηγικής, παίρνουμε ότι τα τιμήματα της αναρχίας και της σταθερότητας είναι πολύ χαμηλά, περίπου 1.06.

Μελετούμε επίσης την κλάση των παιγνίων του μοντέλου conveyor belt. Αυτά είναι πιο περίπλοκα παιχνίδια και εδώ έχουμε εξετάσει μόνο δύο παίκτες και αυθαίρετες συναρτήσεις καθυστέρησης (για δύο παίκτες η τάξη των αφφινε και η τάξη αυθαίρετων συναρτήσεων καθυστέρησης είναι ταυτόσημες). Χαρακτηρίζουν τις ισορροπίες Nash, τη βέλτιστη λύση, και υπολογίζουμε το PoA και PoS. Συγκεκριμένα, δείχνουμε ότι υπάρχει

μια μοναδική συμμετρική Ισορροπία Nash για το μοντέλο conveyor belt στην οποία οι παίκτες εκχωρούν μη μηδενική πιθανότητα σε πολλαπλάσια του $\ell_e(1)$ και αυτές οι πιθανότητες μειώνονται γραμμικά. Επίσης, υπολογίζουμε τη βέλτιστη συμμετρική λύση και υπολογίζουμε το τίμημα της αναρχίας, που κι εδώ προκύπτει ότι είναι 1.06 αλλά υπό διαφορετικές προϋποθέσεις από ότι στην προηγούμενη περίπτωση.

Παιγνιοθεωρητική μελέτη του Παγκόσμιου Ιστού

Στη συνέχεια μελετάμε τον παγκόσμιο ιστό, θεωρώντας τον ως το αποτέλεσμα ενός παιγνίου μεταξύ των συγγραφέων ιστοσελίδων που τοποθετούν υπερσυνδέσμους στη σελίδα τους, στοχεύοντας στη μεγιστοποίηση της φήμης της. Εστιάζουμε στη μελέτη της τοποθέτησης διαφημιστικών συνδέσμων, που συνεπάγονται κάποιο κόστος για τη σελίδα που τους δέχεται, η οποία αποτελεί συνήθη στρατηγική στη βελτιστοποίηση ιστοσελίδων ώστε να αυξάνει η ορατήτητά τους από τις μηχανές αναζήτησης. Αποδεικνύουμε πως ο υπολογισμός της στρατηγικής βέλτιστης απόκρισης των παιχτών είναι NP-hard και υπολογίζουμε μια προσεγγιστική βέλτιστη απόκριση.

Συμπαγής αναπαράσταση δικτύων

Έπειτα προχωρούμε με το σημαντικό τεχνικό θέμα της αναπαράστασης τέτοιων δικτύων ώστε να καταστεί δυνατή η απεικόνισή τους στην κύρια μνήμη. Παρατηρούμε πως οι διάφορες κατηγορίες δικτύων που μας ενδιαφέρουν, όπως για παράδειγμα δίκτυα δρομολόγησης, πληροφόρησης και τα κοινωνικά δίκτυα, τα οποία παραδοσιακά αναπαριστώνται ως γράφοι, έχουν μία κοινή ιδιότητα: επιδεικνύουν υψηλή συγκέντρωση ακμών γύρω από την κεντρική διαγώνιο του πίνακα γειτνίασης του γράφου, πιθανώς μετά από κάποια αναδιάταξη των κόμβων του γράφου. Ως εκ τούτου απομονώνουμε το πυκνό κομμάτι του γράφου και σχεδιάσουμε έναν υβριδικό αλγόριθμο συμπίεσης που μεταχειρίζεται το πυκνό κομμάτι διαφορετικά από ό,τι τον υπόλοιπο γράφο. Παρέχουμε ανάλυση και πειραματική αξιολόγηση της μεθόδου σε ένα αληθινό σύνολο δεδομένων, που δείχνουν πως ξεπερνά σε επιδόσεις την τρέχουσα καλύτερη μέθοδο επιτυγχάνοντας καλύτερο λόγο συμπίεσης και χρόνο προσπέλασης των στοιχείων του γράφου.

Contents

Figures

Tables

Part I

Background and Overview

Chapter 1

Introduction

1.1 This Thesis in a Nutshell

During the last fifteen years, the explosive growth of the internet and its use in everyday life has given rise to various types of networks created by human activity, without any central design or control. The most well known example is the worldwide web (WWW), a service built over the internet that allows users to access information from computers that can be located anywhere in the world, using an extremely simple interface. Of course, the WWW was initiated many years earlier, around the mid-80s, but it became an object of study when it started being used by a huge number of users and thus attracted a lot of information-oriented and economic activity. More recently a number of other networks available through internet have been developed, that also offer abstractions of the social context of the users, referred to as social networks. The development of the internet has also motivated the study of routing networks, as there are usually various alternative ways for the communication between two network entities.

The popularity of such networks has clearly affected the incentives of their users. For instance, although a web page author's main thought when deciding the hyperlinks that would connect her page to the rest of the WWW was to build a page of *high quality content*, a few years later the *popularity* of her page, which can be easily transformed into economic benefit, became her main concern. However, attracting users is not desirable in all types of networks. For example, in routing networks it is usually in a user's interest to *avoid* crowded communication channels. Apparently, since we need to be able to understand the mechanisms behind the creation of such networks, network analysis has become an intriguing object of study.

Apart from the user point of view consequences of the networks' popularity, there are also consequences from the system point of view. Perhaps the most basic such issue is that the network representations cannot fit in a computer's main memory, leading to two drawbacks: a restriction on the network size that can be handled, or poor performance of the critical applications that run over such network infrastruc-

tures, implied by keeping the representation in secondary memory. Neither of them is acceptable, so we need compressed representations. However, compression may well turn the retrieval of network elements to a complicated, thus time consuming process. So we need compressed network representations that allow fast access to the network's elements.

This thesis has been conducted along the two main directions sketched above. The first one is the study of *selfish behaviour* in various networks of special interest in our days, in particular routing networks and information networks. In both types the users are trying to maximize their own payoff from the network; the different incentives, however, make the individual study of each network type necessary. In the case of routing networks, which we study theoretically using congestion games, we introduce and study the class of congestion games with time-dependent strategies. On a more practical side, we model and study the worldwide web, which is the most important information network today, from a game-theoretic point of view.

The second direction has to do with an important technical issue, the representation of these network structures and more specifically with their *compact yet efficient representation*, that can boost the performance of critical applications that run on such networks. We study the common properties of the networks of interest and exploit them to design an efficient compression method.

1.2 Contributions of this Thesis

The contribution of this thesis is threefold:

- we introduce the time dimension in congestion games, proposing a class of games in which players do not only decide the set of resources they are going to use but also they time they enter the game,
- we study the game-theoretic aspects of link placement in the worldwide web,
- we improve the state-of-the-art graph compression algorithm for network graphs created by human activity.

1.3 Tips for Reading this Thesis

1.3.1 Prerequisites

The main prerequisite for this thesis is a facility with the basic notions and techniques for the computation of Nash equilibria and their efficiency. A survey of these concepts

by Nisan et al. [103], intended for computer scientists, is particularly suited for appreciation of the results in this thesis. Additional details can be found in a standard reference on game theory such as Osborne and Rubinstein [108]. We also assume some familiarity with the basics of the theory of NP-completeness, for which Garey and Johnson [67] and Arora and Barak [14] are standard references.

1.3.2 Outline

Chapter 2 presents the technical and conceptual background underlying our results. The remainder of this thesis splits our contributions in two parts, according to the context of the problems studied.

In Part II we study the effect of selfish behaviour in two types of networks. We first consider routing (or traffic) networks, which are traditionally modeled as congestion games and study the effect of timing the participation in such games. The computational model and the study of such networks are presented in Chapter 3. We then consider information networks, focusing on the most well known representative of this class, the worldwide web, and study its game-theoretic aspects. In Chapter 4, we study the establishment of reference and advertising hyperlinks to web pages and their effect to the ranking of web pages.

Part III presents our work on representing routing and information networks. In Chapter 5 we deal with the issue of designing compact yet efficient representations of such networks, proposing an algorithm that improves over the state-of-the-art. We present the analysis of the algorithm and its experimental evaluation based on a dataset of web graphs, road and social networks.

In Part IV (Chapter 6) we outline the conclusions of this thesis and give directions for future research on the above topics.

1.3.3 Dependencies

Chapter 2 is a prerequisite for Part II, as it provides a background in the notation and techniques that are going to be used throughout the text. The sections of Chapter 3 should be read in order.

1.4 Bibliographic Notes

Most results presented in this thesis appear in one of the following previously published, or under submission, works:

- Elias Koutsoupias and Katia Papakonstantinopoulou. **Contention issues in congestion games**. In *Proceedings of the 39th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 623–635, Warwick, UK, July 2012.

- Panagiotis Liakos, Katia Papakonstantinopoulou and Michael Sioutis. **Pushing the Envelope in Graph Compression**. In *23rd ACM International Conference on Information and Knowledge Management (CIKM 2014)*, Shanghai, China, November 2014.

- Panagiotis Liakos, Katia Papakonstantinopoulou and Michael Sioutis. **On the Effect of Locality in Compressing Social Networks**. In *Advances in Information Retrieval - 36th European Conference on IR Research (ECIR)*, pages 650–655, Amsterdam, The Netherlands, April 2014.

- Panagiotis Liakos, Katia Papakonstantinopoulou and Michael Sioutis. **SiVaC*: An Efficient Graph Compression Algorithm**. *Technical report*, University of Athens, Greece, February 2013. Poster presented at WSDM '13. 1st place in Data Challenge organized by the conference.

- Panagiotis Liakos, Katia Papakonstantinopoulou and Michael Sioutis. **A Simple Algorithm for Compressing Web-like Graphs Efficiently**. *Technical report*, University of Athens, Greece, August 2013.

- Elias Koutsoupias and Katia Papakonstantinopoulou. **Game-theoretic aspects of link placement in the worldwide web**. Under preparation.

Chapter 2

Preliminaries

2.1 Notation

2.1.1 Games

A game in normal form, or normal-form game, has $r \geq 2$ players, $1, ..., r$, and for each player $p \leq r$ a finite set S_p of pure strategies. The set S of pure strategy profiles is the Cartesian product of the S_p's. We denote the set of pure strategy profiles of all players other than p by S_{-p}.

A mixed strategy for player p is a distribution on S_p, that is, real numbers $x_j^p \geq 0$ for each strategy $j \in S_p$ such that $\sum_{P_j \in S_p} x_j^p = 1$.

The set (strategy vector) $s \in S$, where $S = \times_p S_p$, selected by the players determines the outcome for each player, which can be expressed through the player's ulitity u_p in order to quantify every player's preference ordering on all possible outcomes.

2.1.2 Congestion Games

The class of congestion games is identical to the class of potential games introduced by Rosenthal in 1973. Rosenthal proved that any congestion game is a potential game and Monderer and Shapley (1996) proved the converse: for any potential game, there is a congestion game with the same potential function.

The class of congestion games consists of atomic and non atomic games, depending of whether the number of users is finite or not. In this work we consider atomic (discrete) congestion games.

Definition 1 (Congestion game)**.** A congestion model $(N, M, (A_i)_{i \in N}, (c_j)_{j \in M})$ is defined as follows:

- $N = \{1, \ldots, n\}$ denotes the set of players
- $M = \{1, \ldots, m\}$ denotes the set of facilities

- For $i \in N$, A_i denotes the set of strategies of player i, where each $a_i \in A_i$ is a non empty subset of the facilities
- For $j \in M, c_j \in R^n$ denotes the vector of costs, where $c_j(k)$ is the cost related to each user of facility j, if there are exactly k players using that facility.

The *congestion game* associated with a congestion model is a game in strategic form with the set of N players, with sets of strategies $(A_i)_{i \in N}$ and with cost function defined as follows: Let $A = \times_{i \in N} A_i$ be the set of all possible deterministic profiles (players' strategy vectors). For any $\vec{a} \in A$ and for any $j \in M$, let $n_j(\vec{a})$ be the number of players using facility j, assuming $\vec{a}$ to be the current profile.

The overall cost function for player i is defined as: $u_i(\vec{a}) = \sum_{j \in a_i} c_j(n_j(\vec{a}))$.

Remark. All players are equal in a sense that they have the same 'weight' (it doesn't matter *which* players are using a facility, only *how many* players are using it).

A discrete congestion game consists of the following components:

- A base set of congestible elements E;
- n players;
- A finite set of strategies S_i for each player, where each strategy $P \in S_i$ is a subset of E;
- For each element e and a vector of strategies $(P_1, P_2, \ldots, P_n)$, a load $x_e = \#\{i : e \in P_i\}$;
- For each element e, a delay function $d_e : \mathbb{N} \longrightarrow \mathbb{R}$;
- Given a strategy P_i, player i experiences delay $\sum_{e \in P_i} d_e(x_e)$. Assume that each d_e are positive and monotone increasing.

We study time-dependent strategies for atomic congestion games and the effect of malicious behavior in atomic congestion games.

2.1.3 Web Graph

The worldwide web is usually represented by a graph, whose nodes and edges correspond to web pages and hyperlinks respectively. The www has specific properties that are outlined in Chapter 4, and we need www models, that is, algorithms that generate www-like graphs, in order to be able to predict the evolution of the www and improve the algorithms that run over it.

2.2 Nash Equilibria Basics

A strategy vector $s \in S$ is said to be a *pure Nash equilibrium* if for all players p and each alternate strategy $s'_p \in S_p$ we have:

$$u_p(s_p, s_{-p}) \geq u_p(s'_p, s_{-p}),$$

in other words, given the other players' strategies, player p has no incentive to change from s_p to s'_p, as its utility will not increase.

Analogously, the *mixed Nash equilibrium* is defined for mixed strategies, i.e. given all others' mixed strategies, a player p has no incentive to alter $x^p_j, \forall j \in S_p$.

2.3 Efficiency of Nash Equilibria

A system has multiple Nash equilibria in general. These equilibria may differ in the total cost induced for the players. We are usually interested in the worst case scenario of the system under consideration, thus the Nash equilibrium with the highest cost is of specific importance. In order to quantify the effect of selfishness on a system we use the Price of Anarchy (or coordination ratio) which is the ratio of the worst case Nash equilibrium over the minimum possible cost of the system (achieved by a configuration designed centrally, ignoring the incentives of the players).

Part II

Selfish Behavior in Routing and Information Networks

Chapter 3

Timing the Participation in Routing

3.1 Introduction

In the last dozen years, the concepts of the price of anarchy (PoA) and stability (PoS) have been successfully applied to many classes of games, most notably to congestion games and its relatives [91, 115, 103]. In congestion games, the players compete for a set of resources, such as facilities or links; the cost of each player depends on the number of players using the same resources; the assumption is that each resource can be shared among the players, but with a cost. Another interesting class of games are the contention games [66] in which the players again compete for resources, but the resources cannot be shared. If more than one players attempt to share a resource at the same time, the resource becomes unavailable and the players have to try again later. There are however interesting games that lie between the two extreme cases of the congestion and contention games. For example, the game that users play for dealing with congestion on a network seems to lie in between—the TCP congestion control policy is a strategy of this game. Timing is part of the strategy of the players (as in contention games) and the latency of a path depends on how many players use its edges (as in congestion games).

In this work, we attempt to abstract away the essential features of these games, to model them, and to study their properties, their Nash equilibria, and their price of anarchy and stability. The games that we consider are essentially congestion games with the addition of time dimension. The difference with congestion games is that players now don't simply select which path to use, but they also decide *when* to initiate the transmission.

Consider a link or facility e of a congestion game with latency function ℓ_e. In the congestion game the latency that a player experiences on the link is $\ell_e(k)$, where k is the number of players that use the link. In our model however, in which the players can also decide when to start, the latency needs to be redefined. We define and study two latency models for the links:

The boat model: in which only the group of players that start together affect the latency of the group: imagine that one boat departs from the source of the link at every time step; all players that decide to start at time t enter the boat which takes them to their destination; the speed of the boat depends only on the number of players in the boat and it is independent of the players on the other boats.

The conveyor belt model: in which the latency of a player depends on the number of other players using the link at the same time regardless if they started earlier or later. Specifically, the link is like a conveyor belt from the source to the destination; the speed of the belt at every time depends on the number of people on it. An interesting variant of this model is when the player is affected only by the players that have been already in the link but not by the players that follow; we don't study this model in this work.

Notice that in the boat model, the order in which the players finish a link may differ from the order in which they start. This, for example, can happen when a player starts later but with a smaller group of people. This cannot happen in the conveyor belt model.

In this work, we consider

- non-adaptive strategies, in which the players decide on their strategy in advance. Their pure strategies consist of a path and a starting time.
- symmetric strategies

Intuitively, in the boat model, the aim of the players is to select a path with small latency and to avoid other players that start at the same time. In the conveyor belt model the aim is similar but the players try to avoid other players that start *near* the same time.

3.1.1 Summary of Results and Techniques

We first study structural properties of the boat and conveyor belt games. We establish that the boat games are congestion games; in contrast, we give examples that show that conveyor belt models are not in general congestion games with the exception of the case of two players. In fact, even simple games with 3 players may not even possess pure Nash equilibria.

In the next section, we characterize the symmetric Nash equilibria of the boat model game for parallel links of affine latency functions, i.e., $\ell_e(k) = a_e\, k + b_e$, and any number of players. We show that there is a unique symmetric mixed Nash equilibrium for these

games. At the Nash equilibrium the probability that a player starts at time t drops linearly on t.

We also compute the optimal symmetric solution. Interestingly, in both the boat and conveyor belt model, the optimal symmetric strategy has exactly the same form with the Nash equilibria but it is less aggressive. That is, in the optimal symmetric strategy the probabilities drop also linearly in time but they are spread out to more strategies. The optimal strategy is a Nash equilibrium of a game with higher latency functions (by almost a factor of 2). A similar bicriteria relation between the Nash equilibria and the optimal solution has been observed in simple congestion games before [115].

From the characterization of the Nash equilibria and the optimal strategy, we get that the price of anarchy and stability is very low $3\sqrt{2}/4 \approx 1.06$. This is the price of anarchy (and stability) when we fix the latencies and let the number of players tend to infinity; when the latency function is tailored to the number of players n, the price of anarchy can be as high as $8n/(7n+1)$.

We also study the class of conveyor belt games. These are more complicated games and here we consider only two players and arbitrary latency functions (for two players the class of affine and the class of arbitrary latency functions are identical). We again characterize the Nash equilibria, the optimal solution, and we compute the PoA and the PoS. Specifically, we show that there exists a unique symmetric Nash equilibrium for the conveyor belt model in which the players assign non-zero probabilities to multiples of $\ell_e(1)$ and these probabilities drop linearly. The explanation of the nature of these equilibria is this: a player attempts with some probability to start at some time $t = 0$; the probability has to balance the risk of the other player starting also at time $t = 0$ and the delay incurred by waiting. The interesting property of the Nash equilibrium is that the player waits enough time steps in order to avoid interference with the other player, had he started at time $t = 0$. After exactly $\ell_e(1)$ steps, with some probability, the player attempts again and the process is repeated.

The price of anarchy and stability is (for large latencies) again approximately $3\sqrt{2}/4 \approx 1.06$. This is the price of anarchy we computed for the boat model, but the relation is not as straightforward as it may appear: in the boat model we take the limit as the number of players tends to infinity, while in the conveyor model, we take the limit as the latencies tend to infinity. In fact, the latter is the same limit as keeping the latencies steady and letting the time step to tend to 0 (thus approximating a continuous-time protocol).

To our knowledge, these games differ significantly from the classes of congestion games that have been studied before. Also, the techniques developed for bounding the PoA and the PoS for congestion games do not seem to be applicable in our setting.

In particular, the smoothness analysis arguments [46, 114, 47] do not seem to apply because we consider symmetric equilibria. In fact, the focus and difficulty of our analysis is to characterize the Nash equilibria and not to bound the PoA (or PoS).

The decision to study only symmetric strategies is based on the assumption that these games are played by many players with no coordination among them. We consider this work as a step towards the study of real-life situations such as the TCP congestion control mechanism in which the players are essentially indistinguishable and therefore symmetric.

In all the games that we study, there exists a unique symmetric equilibrium. For this type of equilibria, the definition of the price of anarchy is uncomplicated: We simply take the ratio of the cost of one player over the cost of one player *of the symmetric optimal solution*. Since there is a unique Nash equilibrium, the price of stability is equal to the price of anarchy.

3.1.2 Related Work

Contention resolution in communication networks is a problem that has attracted the interest of diverse communities of Computer Science. Its significance comes from the fact that contention is inherent in many critical network applications. One of them is the design of multiple access protocols for communication networks, such as Slotted Aloha: According to it, a source transmits a packet through the network, as soon as this packet is available. If a collision takes place, that is, another source attempted to transmit simultaneously, the source waits for some random number of time slots and attempts to retransmit at the beginning of the next slot. The increase of users of the network incurs a large number of collisions and subsequently poor utilization of the system's resources.

During the last four decades many more refined multiple access protocols have been proposed to increase the efficiency of Aloha, the vast majority of which assume that the agents follow the protocol, even if they might prefer not doing so. Recently, slotted Aloha has been studied from a game-theoretic point of view, trying to capture the selfish nature of its users. Part of this work has been done by Altman et al. [7, 8]. The authors model slotted Aloha as a game among the transmitters who aim at transmitting their stochastic flow, using the retransmission probability that maximizes their throughput [7] or minimizes their delay [8]. They show that the system possesses symmetric equilibria and that its throughput deteriorates with larger number of players or arrival rate of new packets. Things get better considering a cost for each transmission though. Another slotted Aloha game is studied by MacKenzie and Wicker [96]. Here the agents aim at minimizing the time spent for unsuccessful transmissions before each

successful one, while each transmission incurs some cost to the player. Their game possesses a symmetric equilibrium, and some of its instantiations possess equilibria that achieve the maximum possible throughput of Aloha.

Much of the prior game-theoretic work considers transmission protocols that always transmit with the same fixed probability. In [66] and [49] the authors consider more complex protocols (multi-round games), where a player's transmission probability is allowed to be an arbitrary function of his play history and the sequence of feedback he has received, and propose asymptotically optimal protocols. In [66], the authors propose a protocol which is a Nash equilibrium and has constant price of stability, i.e., all agents will successfully transmit within time proportional to their number. This protocol assumes that the cost of any single transmission is zero. In [49] the case of non-zero transmission cost is addressed, and a protocol is proposed where after each time slot, the number of attempted transmissions is returned as feedback to the users.

There is a lot of work on game theoretic issues of packet switching. For example, [82] considers the game in which users select their transmission rate, [4] considers TCP-like games in which the strategies of the players are the parameters of the AIMD (additive increase / multiplicative decrease) algorithm, and [68] considers game-theoretic issues of congestion control. All these works are concerned with the steady or long term version of the problems and they don't consider time-dependent strategies in the spirit of this work.

Routing in networks by selfish agents is another area that has been extensively studied based on the notion of the price of anarchy (PoA) [91] and the price of stability (PoS) [10]. The PoA and the PoS compare the social cost of the worst-case and best-case equilibrium to the social optimum. Selfish routing is naturally modeled as a congestion game. The class of congestion or potential games [113, 101] consists of the games where the cost of each player depends on the resources he uses and the number of players using each resource. The effect of selfishness in infinite congestion games was first studied in [115] and of finite congestion games in [46, 20].

The above results concern classical networks or static flows on networks. Perhaps the closest in spirit to our work are the recent attempts to study game-theoretic issues of dynamic flows, or more precisely, of flows over time. In [87], the authors consider selfish selection of routing paths when users have to wait in a FIFO queue before using every edge of their paths; the waiting time is not part of their strategy, but depends on the traffic in front of them. The same model is assumed by [97] who considers the Braess' paradox for flows over time. More results appeared in [28] which gives an efficiently computable Stackelberg strategy for which the competitive equilibrium is not much worse than the optimal, for two natural measures of optimality: total

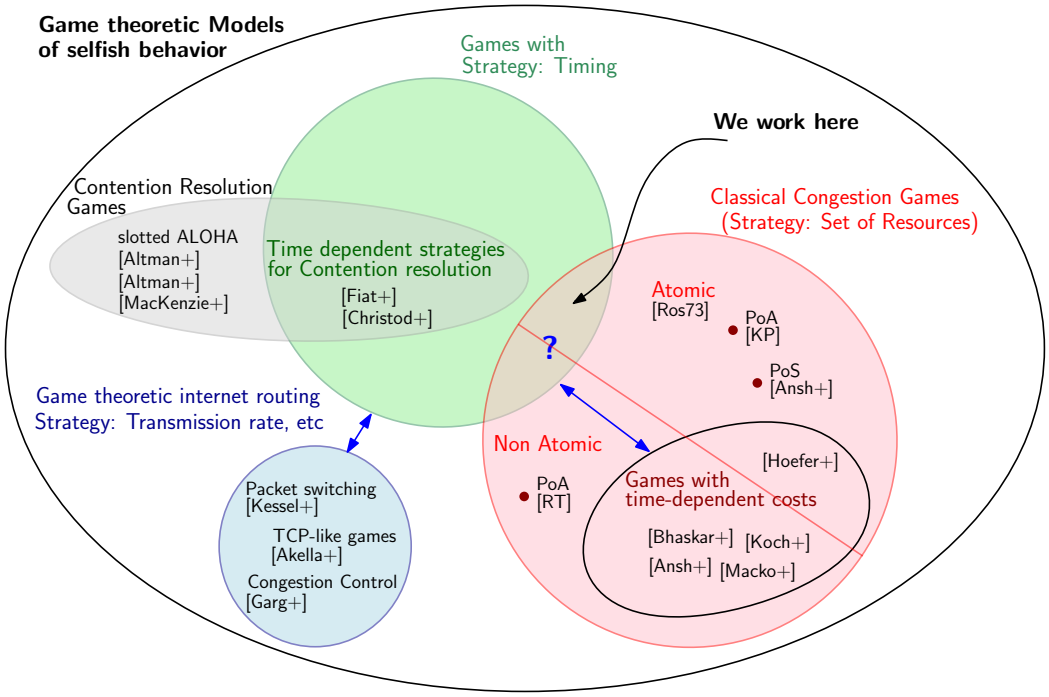


Figure 3.1: Related work

delay of the players and time taken to route a fixed amount of flow to the sink. In a slightly different model, [12] considers game-theoretic issues of discrete-time models in which the latency of each edge depends on its history. All these papers consider non-atomic congestion games. In a different direction which involves atomic games, [71] considers temporal congestion games that are based on coordination mechanisms [48] and congestion games with time-dependent costs.

All these models share with this work the interest in game-theoretic issues of timing in routing, but they differ in an essential ingredient: in our games, timing is the most important part of the players strategy, while in the previous work, time delays exist because of the interaction of the players; in particular, *in all these models the strategy of the players is to select only a path*, while in our games the strategy is essentially the timing. We view our model as a step towards understanding games related to TCP congestion control; this does not seem to be in the research agenda of game-theoretic issues of flows over time.

Figure 3.1 illustrates the aforementioned areas and the relations among them.

3.1.3 Organization

In the following sections we present our results regarding the boat and conveyor belt models. In section 3.2 we show the structural properties of the two models. Then, in section 3.3 we focus on the boat model and compute its Nash equilibria (in 3.3.1) and price of anarchy (in 3.3.3). A similar presentation follows for the conveyor belt model in section 3.4. We compute its Nash equilibria and price of anarchy in 3.4.1 and 3.4.3 respectively.

3.2 Structural Properties of the *Boat* and *Conveyor Belt* Models

Formally, the games that we study here are the following: Let G be a *network* congestion game with n players and latency functions $\ell_e(k)$ on its link e. We define two new games based on G, the boat model game and the conveyor belt game. The pure strategies of both new games of every player consist of one strategy (path) of the original game and one non-negative time step $t \in \mathbb{Z}_0^+$. Their difference lies in the cost of the pure strategies.

In the boat model, the cost of a player is simply $t + \sum_{e \in P} \ell_e(n_t(e))$, where $n_t(e)$ denotes the set of players that also start at time t and use edge e. In the conveyor belt model the cost is more complicated. It depends on the notion of work: in a time interval $[t, t + \Delta t]$ in which player i uses link e, it completes work $\Delta t / \ell_e(k)$, where k is the number of players using the same link during this time interval. A player finishes a link when it completes total work of 1 for this link; the player then moves to the next link of its path.

The boat model The boat model is much simpler than the conveyor belt model. In fact, it is easy to see that the games in the boat model are congestion games: Consider a congestion game G with latency functions $\ell_e(k)$ on its edge (or more generally facility) e. To get the associated game for the boat model we create copies G_0, G_1, ... In the boat model, each player can now play on G_0 immediately, on G_1 after 1 time-step and, in general, on G_t after t time-steps. Equivalently, the associated edge e of G_t has latency function $t + \ell_e(k)$.

The conveyor belt model The definition of the games of the conveyor belt model, albeit intuitively clear, does not allow for a simple expression of the cost as in the case of the boat model.

An example would be more illuminating: consider a link e and two players that start using e at times t_1 and t_2 with $t_1 \leq t_2$. Let f_1 and f_2 denote their finish times. Assuming that the players overlap, or equivalently $t_2 \leq t_1 + \ell_e(1)$, their finish times are given by the linear system:

$$\frac{t_2 - t_1}{\ell_e(1)} + \frac{f_1 - t_2}{\ell_e(2)} = 1$$
$$\frac{f_1 - t_2}{\ell_e(2)} + \frac{f_2 - f_1}{\ell_e(1)} = 1$$

If the players do not overlap on e, their finish times are simply $f_i = t_i + \ell_e(1)$. By solving the above system we can express the finish times as

$$f_i = t_i + \ell_e(1) + \max\left(0, (\ell_e(2) - \ell_e(1))\left(1 - \frac{|t_2 - t_1|}{\ell_e(1)}\right)\right) \tag{3.1}$$

We can use the above approach to find the finish times for three or more players, but trying to express them as in (3.1) does not seem to be useful.

In fact for 2 players, it is easy to compute the finish times for every network. The basic reason for this is that in every edge the difference of their finish times is equal to the difference of their start times: If t_1 and t_2 are the start times on the edge, then the finish times, as given by (3.1), satisfy $f_2 - f_1 = t_2 - t_1$ (independently of whether they overlap or not). Specifically, let (P_i, t_i) be a strategy of player i (i.e., he selects to use path P_i and to start at time t_i). Let us consider the latency of player i at some edge $e \in P_i$. If $t_{e,i}$ denotes the time player i starts using edge e, the latency of edge e is $\ell_e(1) + \max\left\{0,\ (\ell_e(2) - \ell_e(1)) \cdot \left(1 - \frac{|t_{e,2} - t_{e,1}|}{\ell_e(1)}\right)\right\}$. To compute it, we need to know the difference in start times $t_{e,2} - t_{e,1}$. But since the difference is preserved in edges used by both players, the difference is determined by the latency functions of the parts of the paths P_i before edge e:

$$t_{e,2} - t_{e,1} = t_2 - t_1 + \sum_{\substack{e' \in P_2 \\ e' \text{ appears before } e \text{ in } P_2}} \ell_{e'}(1) - \sum_{\substack{e' \in P_1 \\ e' \text{ appears before } e \text{ in } P_1}} \ell_{e'}(1). \tag{3.2}$$

We now turn our attention to structural properties of the conveyor belt games. Unlike the boat games, they are not in general congestion (or potential) games. To show this, it suffices to show that they have no pure equilibrium. This is sufficient, because all congestion games have at least one pure Nash equilibrium [113].

Lemma 1. *There are conveyor belt games for a single link and 3 players that have no pure equilibria.*

Proof. Consider the game with latency function $\ell_e(k) = 5k - 1$. We will show that it

has no pure equilibrium by contradiction. Suppose that there is a pure (symmetric or not) Nash equilibrium in which the players start at times $t_1 \leq t_2 \leq t_3$. Clearly if this is a Nash equilibrium, we must have $t_1 = 0$. Let f_1, f_2, f_3 be the finish times of the 3 players.

Case 1: Players 1 and 3 do not overlap ($f_1 \leq t_3$). The finish time f_1 of the first player satisfies

$$\frac{t_2}{\ell_e(1)} + \frac{f_1 - t_2}{\ell_e(2)} = 1,$$

or equivalently, $f_1 = 9 - 5t_2/4$. We now consider player 2. By time t_3, player 2 would have completed work

$$\frac{f_1 - t_2}{\ell_e(2)} + \frac{t_3 - f_1}{\ell_e(1)} = \frac{1}{16}t_2 + \frac{1}{4}t_3 - \frac{5}{4}.$$

This is increasing in t_2, and therefore player 2 would select t_2 as large as possible: $t_2 = \ell_e(1) = 4$. Now the best strategy for player 3 is to select $t_3 = 0$ and finish at $f_3 = 106/9 < 12$, a contradiction since we assumed that $t_3 \geq t_2$.

Case 2: Players 1 and 3 overlap ($t_3 \leq f_1$). Then, given the starting times $t_1 = 0$, t_2, t_3, the finish times are computed by the following system:

$$\begin{aligned}
\frac{t_2 - t_1}{\ell_e(1)} + \frac{t_3 - t_2}{\ell_e(2)} + \frac{f_1 - t_3}{\ell_e(3)} &= 1 \\
\frac{t_3 - t_2}{\ell_e(2)} + \frac{f_1 - t_3}{\ell_e(3)} + \frac{f_2 - f_1}{\ell_e(2)} &= 1 \\
\frac{f_1 - t_3}{\ell_e(3)} + \frac{f_2 - f_1}{\ell_e(2)} + \frac{f_3 - f_2}{\ell_e(1)} &= 1
\end{aligned}$$

Solving it, we get that $f_3 = 14 - \frac{5}{36}t_2 - \frac{1}{9}t_3$. We observe that the best strategy for player 3 is to set t_3 as large as possible, that is, to value $t_3 = f_1$. But then this becomes equivalent to Case 1. □

The example in the proof of the lemma has nonnegative affine latency function, but one of the coefficients is negative. If both coefficients are nonnegative, i.e., $\ell_e(k) = a_e k + b_e$ with $a_e, b_e \geq 0$, then the game has a pure Nash equilibrium: It is not hard to see that when all players start at $t_i = 0$, they have no reason to switch. Since these games have a pure equilibrium, are they congestion games? The answer is negative: For example, the game of 3 players on a single link with $\ell_e(k) = k$ is not a congestion game or equivalently, it does not admit an exact potential [101]. To verify this, consider the cost (latency) of the players when the start times are $T_\emptyset = (0, 0, 0)$, $T_1 = (1, 0, 0)$, $T_2 = (0, 2, 0)$, and $T_{12} = (1, 2, 0)$. It is straightforward to compute the finish times $F_\emptyset = (3, 3, 3)$, $F_1 = (3, 5/2, 5/2)$, $F_2(2, 3, 2)$, and $F_{12} = (2, 3, 1)$. If there exists an exact potential Φ, it must satisfy $\Phi(T) - \Phi(T') = F_i - F'_i$, for every vectors of starting times

T and T' which differ only on player i and for which F and F' are the corresponding vectors of finish times. In particular, this would imply

$$F_{12,2} - F_{1,2} + F_{1,1} - F_{\emptyset,1} = F_{12,1} - F_{2,1} + F_{2,2} - F_{\emptyset,2}.$$

Since this does not hold, it follows that even the simplest conveyor belt game is not a congestion game for 3 or more players. The case of 2 players is an exception as the following lemma establishes.

Lemma 2. *The conveyor belt model games for 2 players are congestion games.*

Proof. To show that the game of two players is a congestion game, it suffices to exhibit a potential, since the classes of exact potential games and of congestion games are identical [113, 101]. Let (P_i, t_i) be a strategy of player i. The latency of player i on some edge $e \in P_i$ depends on the time that the players starts using this edge. Specifically, if $t_{e,i}$ is the time player i starts using edge e, the *latency on edge* e is $\ell_e(1) + \max\left\{0,\, (\ell_e(2) - \ell_e(1)) \cdot \left(1 - \frac{|t_{e,2} - t_{e,1}|}{\ell_e(1)}\right)\right\}$. The crucial fact is that this latency is the same for both players (because the expression is symmetric with respect to $t_{e,1}$ and $t_{e,2}$). The total latency $c_1((P_1, t_1), (P_2, t_2))$ of player 1 is given by

$$c_1(P_1, t_1, P_2, t_2) = t_1 + \sum_{e \in P_1} \ell_e(1) + \sum_{e \in P_1 \cap P_2} \max\left\{0,\, (\ell_e(2) - \ell_e(1)) \cdot \left(1 - \frac{|t_{e,2} - t_{e,1}|}{\ell_e(1)}\right)\right\}$$

and a similar expression holds for player 2. We can therefore define the potential

$$\Phi(P_1, t_1, P_2, t_2) =$$

$$t_1 + t_2 + \sum_{e \in P_1} \ell_e(1) + \sum_{e \in P_2} \ell_e(1) + \sum_{e \in P_1 \cap P_2} \max\left\{0,\, (\ell_e(2) - \ell_e(1)) \cdot \left(1 - \frac{|t_{e,2} - t_{e,1}|}{\ell_e(1)}\right)\right\},$$

where $t_{e,2} - t_{e,1}$ is defined by (3.2). The function Φ is a potential because $\Phi(P_1, t_1, P_2, t_2) - \Phi(P_1', t_1', P_2, t_2) = c_1(P_1, t_1, P_2, t_2) - c_1(P_1', t_1', P_2, t_2)$ and a similar equality holds for player 2.

In particular, for a single link e and 2 players, we can construct a congestion game directly: We create facilities $e_0, e_1, \ldots$ with latency functions

$$\ell'_{e_t}(1) = 1 + \left\lfloor \frac{t}{\ell_e(1)} \right\rfloor$$
$$\ell'_{e_t}(2) = \ell_{e_t}(1) + \frac{\ell_e(2) - \ell_e(1)}{\ell_e(1)}$$

Each player now has to play $\ell_e(1)$ consecutive edges: that is, when the player chooses

to play at time t, the player must participate in all edges e_t,e_{t+1},. . . ,$e_{t+\ell_e(1)-1}$. It is not hard to see that this is indeed the associated conveyor belt game. □

The following theorem summarizes the above results for the nature of the time-dependent games.

Theorem 1. *All boat games are congestion games. In contrast, only the 2-player conveyor belt games are congestion games. Furthermore, the conveyor belt games of 3 or more players do not have pure equilibria in general.*

The conveyor belt model highlights the importance of the sequential use of the facilities in network congestion games. In the typical view of such games, the order of using the facilities is not important. This is best illuminated by the fact that the set of network congestion games is a subset of congestion games whose standard definition does not include any ordering of the facilities (i.e., a path is simply viewed as a collection of edges). If for example, a congestion game is defined on a path, any reordering of the edges of the path corresponds to the same game.

However, in the conveyor belt model, the ordering is important. This is also the reason for defining time-dependent games only for network congestion games. As an example, consider the game on a path of 2 edges with latency functions $\ell_{e_1}(k) = k$ and $\ell_{e_2}(k) = k + 1$. If three players start at times $t_1 = 0$, $t_2 = t_3 = 1$ then they will complete the path at times $f_1 = 3$, $f_2 = f_3 = 6$. But when we inverse the order of the edges, they will complete the path at different times: $f_1 = 4$, $f_2 = f_3 = 13/2$. Even in 2-player games the order is important when there are multiple (not-independent) paths. For example, consider a network with 3 edges $e_1 = (u_1, u_2)$, $e_2 = (u_1, u_2)$, and $e_3 = (u_2, u_3)$ (2 parallel edges followed by a single edge) and latency functions $\ell_{e_1}(k) = k$, $\ell_{e_2}(k) = k + 1$, $\ell_{e_3}(k) = k + 1$. If the two players start at u_1 and they follow different paths, they will reach u_3 at times $7/2$ and $9/2$. If we reverse the network however and the players start at u_3, the latencies will be 4 and 5.

3.3 Nash equilibria of the *Boat* Model

In this section, we first consider symmetric Nash equilibria of n players for the boat model of parallel links. We also compute the optimal non-selfish solution and estimate the PoA.

3.3.1 Nash equilibria computation

A pure strategy for a player is to select a link e and a time t. A mixed strategy is given by probabilities $p_{e,t}$ with $\sum_{e,t} p_{e,t} = 1$: the player uses link e at time step t with probability

$p_{e,t}$. A set of probabilities $p_{e,t}$ is a Nash equilibrium when a player has no incentive to change it to some other values q. To find the Nash equilibria, we first estimate the latency $d_{e,t}$ when the player selects pure strategy (e,t):

$$d_{e,t} = t + \sum_{k=0}^{n-1} \binom{n-1}{k} p_{e,t}^k \, (1-p_{e,t})^{n-1-k} \, \ell_e(k+1). \tag{3.3}$$

Let $d = \min_{e,t} d_{e,t}$ denote the minimum value. Then the probabilities define a symmetric mixed Nash equilibrium if and only if $p_{e,t} > 0$ implies $d = d_{e,t}$.

To find the Nash equilibria, the first crucial step is to show that the probabilities in every link must be non-increasing in t. This is shown by the following lemma which holds for arbitrary latency functions, not only for affine ones:

Lemma 3. *If for every edge e the latencies $\ell_e(k)$ are non-decreasing in k, then every symmetric Nash equilibrium is a non-increasing sequence of probabilities: $p_{e,t} \geq p_{e,t+1}$.*

Proof. First observe that when the latencies are non-decreasing (i.e., when $\ell_e(k+1) \geq \ell_e(k)$ for every k), the expression $d_{e,t} - t$ is non-decreasing in $p_{e,t}$. To verify this, take the derivative of $d_{e,t} - t$ in (3.3) with respect to $p_{e,t}$

$$\begin{aligned}
&\frac{\partial (d_{e,t} - t)}{\partial p_{e,t}} \\
&= \sum_{k=1}^{n-1} \binom{n-1}{k} k \, p_{e,t}^{k-1} \, (1-p_{e,t})^{n-1-k} \, \ell(k+1) - \sum_{k=0}^{n-2} \binom{n-1}{k} (n-1-k) \, p_{e,t}^k \, (1-p_{e,t})^{n-2-k} \, \ell(k+1) \\
&= \sum_{k=1}^{n-1} \left(\binom{n-1}{k} k \, \ell(k+1) - \binom{n-1}{k-1} (n-k) \, \ell(k) \right) p_{e,t}^{k-1} \, (1-p_{e,t})^{n-1-k}.
\end{aligned}$$

This is nonnegative because $\binom{n-1}{k} k = \binom{n-1}{k-1}(n-k)$.

To see now that the sequence $p_{e,t}$ is non-increasing, observe that if we had $p_{e,t+1} > p_{e,t}$ then we would have $d_{e,t+1} - (t+1) \geq d_{e,t} - t$. But the last inequality shows that d_{t+1} is not the minimum value, which would imply that $p_{t+1} = 0$, a contradiction. □

We define the support of the Nash equilibrium to be the set of strategies that have minimum latency: $S_e = \{t : d_{e,t} = d\}$. Alternatively, we could have defined the support to be the set of strategies with on-zero probability at the Nash equilibrium; the two notions are similar but not identical in some cases. Notice the convention $d_{e,h_e+1} > d = d_{e,h_e}$, in the definition of the support. The last lemma shows that the support S_e of every link e is of the from $\{0, \ldots, h_e\}$ for some integer h_e.

We now focus on affine latencies functions, $\ell_e(k) = a_e k + b_e$, for which the cost $d_{e,t}$

in (3.3) takes a simple closed form:

$$d_{e,t} = t + a_e + b_e + (n-1)\, a_e\, p_{e,t}, \tag{3.4}$$

which shows that the probabilities of the Nash equilibria are of the form:

$$p_{e,t} = \begin{cases} \frac{d-a_e-b_e-t}{(n-1)a_e} & \text{for } t \le h_e \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

Observe that at every Nash equilibrium $p_{e,t}$, the non-zero probabilities decrease linearly with t. These probabilities are determined by the cost of each player d and the integers h_e (one for each link). In fact, the parameters h_e are very tightly related with the cost d of each player:

Theorem 2. *There is a unique symmetric Nash equilibrium with support $S_e = \{t : 0 \le t \le h_e = \lfloor d - a_e - b_e \rfloor\}$, where d is the expected cost of every player; its probabilities are given by*

$$p_{e,t} = \begin{cases} \frac{d-a_e-b_e-t}{(n-1)a_e} & \textit{for } t \le d - a_e - b_e \\ 0 & \textit{otherwise} \end{cases}$$

The expected cost $L_{NE} = d$ of every player is the unique solution of the equation

$$\sum_e \frac{(\lfloor d - a_e - b_e \rfloor + 1)\,(2(d - a_e - b_e) - \lfloor d - a_e - b_e \rfloor)}{2(n-1)a_e} = 1. \tag{3.6}$$

Its value is approximately

$$d \approx \frac{\sum_e \frac{a_e+b_e}{2(n-1)a_e} + \sqrt{\left(\sum_e \frac{a_e+b_e}{2(n-1)a_e}\right)^2 + \left(\sum_e \frac{1}{2(n-1)a_e}\right)\left(1 - \sum_e \frac{(a_e+b_e)^2}{2(n-1)a_e}\right)}}{\sum_e \frac{1}{2(n-1)a_e}}, \tag{3.7}$$

and as n tends to infinity this tends to $\sqrt{\frac{2n}{\sum_e a_e^{-1}}}$.

Proof. We can determine h_e from the constraints $p_{e,h_e} \ge 0$ and $d_{e,h_e+1} > 0$; the latter is based on the way we defined the support. Indeed, from Equation (3.5) we get that $d - a_e - b_e \ge h_e$. And from Equation (3.4) for d_{h_e+1}, when we take into account that $p_{e,t+1} = 0$, we get $(h_e + 1) + a_e + b_e > d$, or equivalently $h_e + 1 > d - a_e - b_e$. It follows that $h_e = \lfloor d - a_e - b_e \rfloor$.

To simplify the notation, let's define $\eta_e = d - a_e - b_e$; therefore $h_e = \lfloor \eta_e \rfloor$. We observe that the cost d determines completely the parameters η_e and the probabilities at the Nash equilibrium. To show that there is a unique Nash equilibrium, we need to show

that all the Nash equilibria have the same cost d. To compute d we use the fact that the sum of probabilities is 1. We have

$$\sum_{e,t} p_{e,t} = \sum_e \sum_{t=0}^{h_e} \frac{d - a_e - b_e - t}{(n-1)a_e} = \sum_e \sum_{t=0}^{\lfloor \eta_e \rfloor} \frac{\eta_e - t}{(n-1)a_e} = \sum_e \frac{(\lfloor \eta_e \rfloor + 1)(2\eta_e - \lfloor \eta_e \rfloor)}{2(n-1)a_e} \tag{3.8}$$

It is straightforward to check that the function $(\lfloor x \rfloor + 1)(2x - \lfloor x \rfloor)$ is strictly increasing in x for nonnegative x. Therefore, each term $\frac{(\lfloor \eta_e \rfloor + 1)(2\eta_e - \lfloor \eta_e \rfloor)}{2(n-1)a_e}$ in the last sum is strictly increasing in η_e and consequently in d. The sum of all these terms is also strictly increasing in d because it is the sum of strictly increasing functions (one for each edge e). It follows that the equation $\sum_{e,t} p_{e,t} = 1$ has a unique solution. This unique value d completely determines the parameters h_e and the probabilities of the Nash equilibrium in Equation (3.5).

The above define the Nash equilibrium in terms of the cost d. It remains to determine d. Its value is given by (3.6), which expresses the fact that the sum of probabilities in (3.8) is 1. To solve this equation for d, we observe that

$$x^2 \leq (\lfloor x \rfloor + 1)(2x - \lfloor x \rfloor) \leq (x + 1/2)^2.$$

This means that the solution of the equation

$$\sum_e \frac{(x - a_e - b_e)^2}{2(n-1)a_e} = 1,$$

is very close to d (and in particular $x - 1/2 \leq d \leq x$).

It is straightforward to verify that the solution to the above equation is given by (3.7). As the number n of players tends to ∞, the expression is approximately $1/\sqrt{\sum_e \frac{1}{2(n-1)a_e}}$, which shows that the cost of every player tends to $\sqrt{\frac{2n}{\sum_e a_e^{-1}}}$. □

3.3.2 The optimal setting

Let us now consider the optimal symmetric protocol. With similar reasoning, the expected latency of a player is

$$L_{OPT} = \sum_e \sum_{t=0}^{\infty} p_{e,t} \left(t + \sum_{k=0}^{n-1} \binom{n-1}{k} p_{e,t}^k \, (1 - p_{e,t})^{n-1-k} \, \ell_e(k+1) \right) = \sum_e \sum_{t=0}^{\infty} p_{e,t} \, d_{e,t}$$

We seek the probabilities $p_{e,t}$ with $\sum_{e,t} p_{e,t} = 1$ which minimize the above expression. We again focus on affine latencies. With $\ell_e(k) = ak + b$, the above expression has the

following compact form

$$L_{OPT} = \sum_e \sum_{t=0}^{\infty} p_{e,t}\,(t + a_e + b_e + (n-1)\,a_e\,p_{e,t}).$$

We minimize this subject to $\sum_{e,t} p_{e,t} = 1$. Using a Lagrange multiplier and taking derivatives, we get that the minimum occurs when the probabilities have the form $p_{e,t} = (\lambda - a_e - b_e - t)/(2(n-1)a_e)$, for some constant λ, and $p_{e,t} = 0$ when $\lambda - a_e - b_e - t \leq 0$. This means that they decrease linearly with t until $c_e = \lambda - a_e - b_e$, when they become 0 and they remain 0 from that point on. Thus, the form of the optimal probabilities resembles the form of the Nash equilibrium probabilities; the only difference is that the optimal probabilities drop slower to 0 (the factors are $2(n-1)a_e$ and $(n-1)a_e$ respectively). Taking into account the constant term also we get,

Lemma 4. *The set of probabilities of the optimal solution for latencies $\ell_e(k) = a_e k + b_e$ is a Nash equilibrium for latencies $\ell_e(k) = 2a_e k + (b_e - a_e)$.*

Therefore the probabilities of the optimal solution are:

$$p_{e,t} = \begin{cases} \frac{\lambda - a_e - b_e - t}{2(n-1)a_e} & t \leq h_e^* \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

where $h_e^* = \lfloor \lambda - a_e - b_e \rfloor$, and the value of λ is the unique solution of the equation $\sum_{e,t} p_{e,t} = 1$. Thus, λ is determined by an equation similar to (3.6) (they essentially differ only in the denominator):

$$\sum_e \frac{(\lfloor \lambda - a_e - b_e \rfloor + 1)\,(2(\lambda - a_e - b_e) - \lfloor \lambda - a_e - b_e \rfloor)}{4(n-1)a_e} = 1. \tag{3.10}$$

From the probabilities we can compute L_{OPT}. Observe that the optimal case differs from the Nash equilibrium case of the previous subsection in that the parameters λ and L_{OPT} are distinct (while in the Nash equilibrium case they are identical—equal to d).

As in the case of the Nash equilibrium, it is useful to define $\eta_e^* = \lambda - a_e - b_e$. We

can then compute the optimal latency:

$$\begin{aligned} L_{OPT} &= \sum_e \sum_{t=0}^{\infty} p_{e,t}\,(t + a_e + b_e + (n-1)\,a_e\,p_{e,t}) \\ &= \sum_e \sum_{t=0}^{h_e^*} \frac{\eta_e^* - t}{2(n-1)a_e} \left(t + a_e + b_e + (n-1)a_e \frac{\eta_e^* - t}{2a_e(n-1)} \right) \\ &= \sum_e \frac{(h_e^*+1)\ (6\eta_e^*(\eta_e^*+2a_e) - h_e^*(2h_e^*+6a+1))}{24(n-1)a_e} \end{aligned}$$

To get an approximate estimate as n tends to infinity, we observe that λ is approximately given by

$$\sum_e \frac{\lambda^2}{4(n-1)a_e} \approx 1 \quad \Rightarrow \quad \lambda \approx 2\sqrt{\frac{n}{\sum_e a_e^{-1}}}.$$

From this, we can find an approximate value for L_{OPT}:

$$L_{OPT} \approx \frac{\eta_e^{*3}}{6(n-1)a_e} \approx \sum_e \frac{\lambda^3}{6(n-1)a_e} = \frac{4}{3}\sqrt{\frac{n}{\sum_e a_e^{-1}}}$$

3.3.3 The price of anarchy

Comparing the value of L_{OPT} to the cost d of the Nash equilibrium, we see that the PoA and the PoS of the boat model on parallel links with affine latency functions tends to $\frac{3\sqrt{2}}{4} \approx 1.06$, as the number of players n tends to infinity (while the parameters of the network remain fixed).

Theorem 3. *For every fixed set of parallel links with positive a_e and b_e, the PoA (and PoS) tends to $3\sqrt{2}/4 \approx 1.06$, as the number of players n tends to infinity.*

However, for fixed number of players and because of the integrality of h and h^*, the situation is more complicated. Figure 3.2 shows the PoA for typical values of a_e and n, for one link. The situation is captured by the following theorem:

Theorem 4. *For one link and fixed number of players n, the PoA is maximized when $a_e = 1/(n-1)$ and $b_e = 0$. For these value, the NE is pure ($p_{e,0} = 1$), but the optimal symmetric solution is given by the probabilities $p_{e,0} = 3/4$ and $p_{e,1} = 1/4$. For these values, we get $L_{NE} = d = n/(n-1)$, $L_{OPT} = (7n+1)/(8(n-1))$, and PoA$= 8n/(7n+1)$.*

Proof. To compare the costs L_{NE} and L_{OPT} we first investigate the solutions of the equations (6) and (10) as functions of a_e; since we care about the worst-case PoA, we can safely assume that $b_e = 0$ because $b_e \geq 0$ is added to both the numerator and the denominator of the PoA.

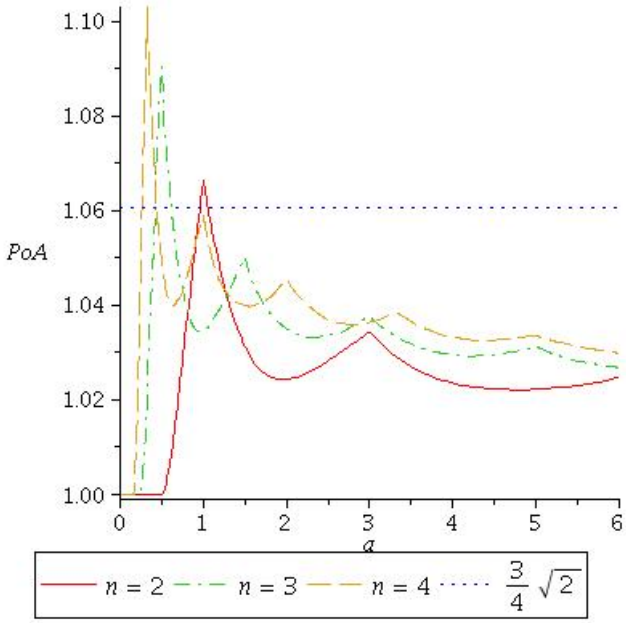


Figure 3.2: PoA of the single-link boat games

For every nonnegative integer k, let us define $A_k = \frac{k(k+1)}{2(n-1)}$ which are the values of a_e where the value $d - a_e$ becomes integral (equal to k). The following lemma gives the solution of (6) for the intervals $[A_k, A_{k+1})$ where the integral part of $d - a_e$ is constant. It also extends it to the optimal cost.

We now compute the values of a_e that maximize the PoA. Both the L_{NE} and L_{OPT} are increasing in a_e. Given a_e and n it is easy to compute k such that $\frac{k(k+1)}{2(n-1)} \leq a_e < \frac{(k+1)(k+2)}{2(n-1)}$; it is $k = \left\lfloor \frac{-1+\sqrt{1+8(n-1)a_e}}{2} \right\rfloor$, and for this k $L_{NE} = \frac{n+k}{k+1}a_e + \frac{k}{2}$. Similarly, we find that $\frac{k^*(k^*+1)}{4(n-1)} \leq a_e < \frac{(k^*+1)(k^*+2)}{4(n-1)}$ for $k^* = \left\lfloor \frac{-1+\sqrt{1+16(n-1)a_e}}{2} \right\rfloor$, and for this k^* $L_{OPT} = \frac{n+k^*}{k^*+1}a_e + \frac{k^*}{2} - \frac{k^*(k^*+1)(k^*+2)}{48(n-1)a_e}$. We now split the possible values of a_e in disjoint intervals, such that in each interval the PoA is given by a single function for all values of a_e, and study each interval separately.

- If $a_e < \frac{A_1}{2} = \frac{1}{2(n-1)}$,
  $L_{NE} = na_e$, $L_{OPT} = na_e$ and thus PoA=1.

- If $\frac{A_1}{2} \leq a_e < A_1 \Leftrightarrow \frac{1}{2(n-1)} \leq a_e < \frac{1}{n-1}$,
  $L_{NE} = na_e$, $L_{OPT} = \frac{n+1}{2}a_e + \frac{1}{2} - \frac{1}{8(n-1)a_e}$ and thus PoA= $\frac{na_e}{\frac{n+1}{2}a_e + \frac{1}{2} - \frac{1}{8(n-1)a_e}}$.

  For $a_e \geq \frac{1}{2(n-1)}$ the PoA is increasing in a_e. Indeed, $\left(\frac{1}{PoA}\right)' = \frac{1-2a_e(n-1)}{4n(n-1)a_e^3} \leq 0$ and therefore $\frac{1}{PoA}$ is decreasing. The maximum value of the PoA is achieved for $a_e = \frac{1}{n-1}$ and it is PoA= $\frac{\frac{n}{n-1}}{\frac{n+1}{2}\frac{1}{n-1} + \frac{1}{2} - \frac{1}{8}} = \frac{\frac{n}{n-1}}{\frac{4(n+1)+3(n-1)}{8}} = \frac{8n}{7n+1}$

- If $A_1 \leq a_e < \frac{1}{2}A_2 \Leftrightarrow \frac{1}{n-1} \leq a_e < \frac{3}{2(n-1)}$,
  $L_{NE} = \frac{n+1}{2}a_e + \frac{1}{2}$, $L_{OPT} = \frac{n+1}{2}a_e + \frac{1}{2} - \frac{1}{8(n-1)a_e}$ and thus PoA= $\frac{\frac{n+1}{2}a_e+\frac{1}{2}}{\frac{n+1}{2}a_e+\frac{1}{2}-\frac{1}{8(n-1)a_e}}$.

  For $a_e \geq \frac{1}{n-1}$ the PoA is decreasing in a_e. Indeed, $\left(\frac{1}{PoA}\right)' = \frac{2a_e(n-1)+4a_e+1}{4(a_en+a+1)^2(n-1)a_e^2} > 0$ and therefore $\frac{1}{PoA}$ is increasing. The maximum value of the PoA is achieved for $a_e = \frac{1}{n-1}$ as in the case above.

- For $a_e \geq \frac{3}{2(n-1)}$ we use the bounds: $L_{NE} \leq a_e + \frac{-1+\sqrt{8(n-1)a_e+1}}{2}$ and $L_{OPT} \geq a_e + \frac{4}{3}\sqrt{(n-1)a_e} - \frac{1}{2}$, given by lemma 6.

  Then PoA$\leq \frac{a_e+\frac{-1+\sqrt{8(n-1)a_e+1}}{2}}{a_e+\frac{4}{3}\sqrt{(n-1)a_e}-\frac{1}{2}} \leq \frac{\frac{3}{2}-\frac{1}{2}+\frac{\sqrt{8(n-1)a_e+1}}{2}}{\frac{3}{2}-\frac{1}{2}+\frac{4}{3}\sqrt{(n-1)a_e}} = \frac{1+\sqrt{2(n-1)a_e+\frac{1}{4}}}{1+\frac{4}{3}\sqrt{(n-1)a_e}}$. The derivative of the last expression with respect to $(n-1)a$ is negative for $(n-1)a \geq \frac{2}{3}$.

Therefore the PoA is maximized when $a_e = \frac{1}{n-1}$.

□

Lemma 5. *Let* $A_k = \frac{k(k+1)}{2(n-1)}$. *For* $a_e \in [A_k, A_{k+1})$

$$L_{NE} = \frac{n+k}{k+1}a_e + \frac{k}{2}.$$

For $a_e \in [A_k/2, A_{k+1}/2)$

$$L_{OPT} = \frac{n+k}{k+1}a_e + \frac{k}{2} - \frac{k(k+1)(k+2)}{48(n-1)a_e}$$

Proof. We first show that for $a_e \in [A_k, A_{k+1})$ the value of d given by equation (6) satisfies $\lfloor d - a_e \rfloor = k$.

Assume $k = \lfloor d - a_e \rfloor$ and let $x = d - a_e - k$. From (6) for one link we get $\frac{(k+1)(2k+2x-k)}{2(n-1)a_e} = 1 \Leftrightarrow x = \frac{(n-1)a_e}{k+1} - \frac{k}{2}$.

For $a_e \in [A_k, A_{k+1})$ we get $\frac{k(k+1)}{2(n-1)} \leq a_e < \frac{(k+1)(k+2)}{2(n-1)} \Leftrightarrow 0 \leq \frac{(n-1)a_e}{k+1} - \frac{k}{2} < 1 \Leftrightarrow x \in [0,1)$ so $\lfloor d - a_e \rfloor = k$ holds. We thus have $L_{NE} = d = a_e + k + x = a_e + k + \frac{(n-1)a_e}{k+1} - \frac{k}{2} = \frac{n+k}{k+1}a_e + \frac{k}{2}$.

In a similar way we show that for $a_e \in [\frac{A_k}{2}, \frac{A_{k+1}}{2})$ the value of λ given by equation (10) satisfies $\lfloor \lambda - a_e \rfloor = k$. Here $\lambda - a_e = k + x'$ with $x' = \frac{2(n-1)a_e}{k+1} - \frac{k}{2}$. For $a_e \in [\frac{A_k}{2}, \frac{A_{k+1}}{2})$ we get $\frac{k(k+1)}{4(n-1)} \leq a_e < \frac{(k+1)(k+2)}{4(n-1)} \Leftrightarrow 0 \leq \frac{2(n-1)a_e}{k+1} - \frac{k}{2} < 1 \Leftrightarrow x' \in [0,1)$ so $\lfloor \lambda - a_e \rfloor = k$ holds.

We thus have $L_{OPT} = \frac{(\lfloor \lambda - a_e \rfloor + 1)(6(\lambda - a_e)(\lambda - a_e + 2a_e) - \lfloor \lambda - a_e \rfloor (2\lfloor \lambda - a_e \rfloor + 6a_e + 1))}{24(n-1)a_e}$
$= \frac{(k+1)(6(k+x)(k+x+2a_e) - k(2k+6a_e+1))}{24(n-1)a_e} = \frac{n+k}{k+1}a_e + \frac{k}{2} - \frac{k(k+1)(k+2)}{48(n-1)a_e}$.

□

Lemma 6. *For one link and fixed number of players n,*

$$a_e + \sqrt{2}\sqrt{(n-1)a_e} - \frac{1}{2} \leq L_{NE} \leq a_e + \sqrt{2}\sqrt{(n-1)a_e}$$

and

$$a_e + \frac{4}{3}\sqrt{(n-1)a_e} - \frac{1}{2} \leq L_{OPT} \leq a_e + \frac{4}{3}\sqrt{(n-1)a_e}.$$

Proof. Take $a_e \in [A_k, A_{k+1})$. Consider the identity

$$\Big(\frac{x}{2(k+1)} + \frac{k}{2}\Big)^2 - x + k = \frac{(x - k(k+1))(x - (k-1)(k+1))}{4(k+1)^2}.$$

This implies that $\frac{x}{2(k+1)} + \frac{k}{2} < \sqrt{x}$ for $x \in [k(k+1), (k+1)(k+2)]$. Let $x = 2(n-1)a_e$. We get $L_{NE} - a_e \leq \sqrt{2(n-1)a_e}$.

Similarly,

$$\Big(\frac{x}{2(k+1)} + \frac{k}{2} + \frac{1}{2}\Big)^2 - x - \frac{1}{4} = \frac{(x - k(k+1))(x - (k+1)(k+2))}{4(k+1)^2}.$$

For $x \in [k(k+1), (k+1)(k+2)]$ we get $\frac{x}{2(k+1)} + \frac{k+1}{2} \leq \sqrt{x + \frac{1}{4}}$. Then for $x = 2(n-1)a_e$ we get $L_{NE} \leq -\frac{1}{2} + \sqrt{2(n-1)a_e + \frac{1}{4}}$.

□

3.4 Nash Equilibria of the *Conveyor Belt* model

We now turn our attention to the conveyor belt model, which is more complicated than the boat model. In the conveyor belt model each link is like a conveyor belt whose speed depends on the number of players on it. *We only consider the case of 2 players in this section.* The cost $c_e(t, t')$ of a player for pure strategies (e, t) when the other player starts using link e at time step t' is computed using $f_i = t_i + \ell_e(1) + \max\Big(0, (\ell_e(2) - \ell_e(1))\Big(1 - \frac{|t_2 - t_1|}{\ell_e(1)}\Big)\Big)$ where t_i, f_i are the start and finish times of player i respectively.

To simplify the discussion, we assume that $\ell_e(1)$ is an integer; this does not seem to really change the nature of equilibria, except perhaps when $\ell_e(1) < 1$ which does not seem a very interesting case.

3.4.1 Nash equilibria computation

Consider a symmetric Nash equilibrium with probabilities $p_{e,t}$, the same for every player. It is a Nash equilibrium when a player has no incentive to change his probabilities to

different values. To find the Nash equilibria, we first compute the expected cost $d_{e,t}$ of a player when he plays pure strategy (e,t):

$$d_{e,t} = \sum_{t'} c_e(t,t') = t + \ell_e(1) + (\ell_e(2) - \ell_e(1)) \sum_{r=-\ell_e(1)}^{\ell_e(1)} \left(1 - \tfrac{|r|}{\ell_e(1)}\right) p_{e,t+r} \tag{3.11}$$

The probabilities define a symmetric mixed Nash equilibrium when probability $p_{e,t} > 0$ implies $d_{e,t} = d = \min_{e,t} d_{e,t}$.

We are interested in symmetric Nash equilibria, that is equilibria that occur when all players use the same strategies. Let's first establish a very intuitive fact:

Claim 1. *If at the Nash equilibrium, positive probability is allocated to edge e, then $p_{e,0} > 0$.*

Proof. Suppose not. Let t be the minimum time for which $p_{e,t} > 0$. We have

$$d_{e,t} = t + \ell_e(1) + (\ell_e(2) - \ell_e(1)) \sum_{r=0}^{\ell_e(1)} (1 - \frac{r}{\ell_e(1)}) p_{e,t+r} \tag{3.12}$$

and

$$d_{e,0} = \ell_e(1) + (\ell_e(2) - \ell_e(1)) \sum_{r=t}^{\ell_e(1)} (1 - \frac{r}{\ell_e(1)}) p_{e,r}$$

From these, by subtracting and ignoring the last terms of $d_{e,t}$, we get that

$$d_{e,t} - d_{e,0} \geq t + (\ell_e(2) - \ell_e(1)) \sum_{r=t}^{\ell_e(1)} \frac{t}{\ell_e(1)} p_{e,r} \geq t > 0$$

which contradicts the Nash equilibrium property. □

The next lemma shows that the support $S_e = \{t \,:\, d_{e,t} = d\}$ of every mixed Nash equilibrium is of the form $\{0, \ldots, \hat{h}_e\}$ for some $\hat{h}_e$.

Lemma 7. *If for some t there exists $s \geq t$ with $p_{e,t} > 0$, then t is in the support S_e, i.e. $d_{e,t} = d$.*

Proof. By induction on t. The base case follows from $p_{e,0} > 0$ which shows that $t = 0$ is in the support S_e. Suppose that the claim is true for some t; we will prove the statement for $t+1$. Let $s = \min\{u : u \geq t+1 \text{ and } p_{e,u} > 0\}$. By the premise of the lemma, the set $\{u : u \geq t+1 \text{ and } p_{e,u} > 0\}$ is not empty.

We first show that $s \leq t + \ell_e(1)$, by a similar argument for $p_{e,0} > 0$: Indeed, if

$s > t + \ell_e(1)$, we have

$$d_{e,s} - d_{e,t+\ell_e(1)} \geq s - t - \ell_e(1) + (\ell_e(2) - \ell_e(1)) \sum_{r=s}^{t+2\ell_e(1)} \frac{s - t - \ell_e(1)}{\ell_e(1)} p_{e,t} \geq s - t - \ell_e(1)$$

Let us clarify that when $s > t + 2\ell_e(1)$, the sum on the right-hand side is over an empty range. In any case, the right-hand side is strictly greater than 0 which contradicts the Nash equilibrium property.

If $s = t + 1$ then $t + 1$ is in the support of the NE and therefore $d_{e,t+1} = d$. Otherwise, consider the expression

$$d_{e,s} - d_{e,t} - (s - t)(d_{e,t+1} - d_{e,t}) =$$
$$\frac{\ell_e(2) - \ell_e(1)}{\ell_e(1)} \left(- \sum_{r=t+1}^{s-1} 2(s - r) p_{e,r} + \sum_{r=t+\ell_e(1)+1}^{s+\ell_e(1)-1} (s + \ell_e(1) - r) p_{e,r} + \sum_{r=t-\ell_e(1)+1}^{s-\ell_e(1)-1} (s - \ell_e(1) - r) p_{e,r} \right),$$

and notice that the negative terms vanish because the probabilities are 0. We immediately get that $d_{e,s} - d_{e,t} - (s - t)(d_{e,t+1} - d_{e,t}) \geq 0$. Since s is in the support of the NE and $d_{e,t} = d$, we have that $d_{e,s} = d$ and $-(s - t)(d_{e,t+1} - d_{e,t}) \geq 0$. This can only happen if $d_{e,t+1} \leq d_{e,t}$, which together with the condition $d_{e,t+1} \geq d$, shows the desired result: $d_{e,t+1} = d$. □

The previous lemma establishes that the support S_e starts at 0 and is contiguous. With this, we can now determine the exact structure of Nash equilibria.

Theorem 5. *The Nash equilibria of the conveyor belt game of two players in parallel links have probabilities*

$$p_{e,t} = \begin{cases} \frac{d - \ell_e(1) - t}{\ell_e(2) - \ell_e(1)} & t \leq d - \ell_e(1) \text{ and } \frac{t}{\ell_e(1)} \in \mathbb{Z}^+ \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

where d is the expected cost of each player and it is the unique solution of the equation

$$\sum_e \frac{(\lfloor \eta_e \rfloor + 1)(2\eta_e - \lfloor \eta_e \rfloor)}{2^{\frac{\ell_e(2) - \ell_e(1)}{\ell_e(1)}}} = 1, \tag{3.14}$$

where $\eta_e = d/\ell_e(1) - 1$.

Proof. Consider some $0 < t < \hat{h}_e$. Then from the definition of $d_{e,t}$ we can compute $d_{e,t+1} - 2d_{e,t} + d_{e,t-1} = \frac{\ell_e(2) - \ell_e(1)}{\ell_e(1)} (p_{e,t-\ell_e(1)} - 2p_{e,t} + p_{e,t+\ell_e(1)})$. Since for $t \in \{1, \ldots, \hat{h}_e - 1\}$, all $t - 1$, t and $t + 1$ are in the support S_e, we have that $d_{e,t-1} = d_{e,t} = d_{e,t+1}$. In turn,

this gives that the right-hand side is 0 and we get that $p_{e,t+\ell_e(1)} - p_{e,t} = p_{e,t} - p_{e,t-\ell_e(1)}$; this shows that if we consider times that differ by $\ell_e(1)$, the probabilities drop linearly and more specifically that for integers k, x: $p_{e,k\ell_e(1)+x} - p_{e,x} = k(p_{e,x+\ell_e(1)} - p_{e,x})$.

This linearity allows us to conclude that $p_{e,t} = 0$ for every t which is not a multiple of $\ell_e(1)$. To see this consider some $x \in \{1, \ldots, \ell_e(1) - 1\}$ and the sequence $p_{e,x-\ell_e(1)}, p_{e,x}, p_{e,x+\ell_e(1)}, \ldots, p_{e,x+k\ell_e(1)}$. This sequence is linear and starts with a 0 (since $x - \ell_e(1) < 0$) and ends again in 0 (if we take k such that $\hat{h}_e < x + k\ell_e(1) \leq \hat{h}_e + \ell_e(1)$).

The above reasoning does not apply to the value $x = 0$, because $p_{e,t+\ell_e(1)} - p_{e,t} = p_{e,t} - p_{e,t-\ell_e(1)}$ only for $t \in \{1, \ldots, \hat{h}_e - 1\}$. To summarize, the NE with support $\{0, \ldots, \hat{h}_e\}$ have non-zero probabilities only on the multiples of $\ell_e(1)$. This means that either the players start together, or they do not overlap, which is *exactly the property of the boat model*. It follows that for one link, the Nash equilibrium is identical to the Nash equilibrium of the boat game with time step expanded to $\ell_e(1)$. For more than one link, the time steps in each link are different, because $\ell_e(1)$ are different. Nevertheless the analysis of the boat model carries over to the conveyor belt model.

The proof now is essentially the same with the boat model, but with the extra restriction that the time steps are not the same in all links. Since the probabilities are non-zero only at integral multiples of $\ell_e(1)$, the expression (3.12) of the latency $d_{e,t}$ becomes $d_{e,t} = t + \ell_e(1) + (\ell_e(2) - \ell_e(1))p_{e,t}$ when t is an integral multiple of $\ell_e(1)$. It follows that the probabilities are as in (3.13). The cost d is determined by the equation $\sum_{e,t} p_{e,t} = 1$. Using the expressions for the probabilities, this equation is equivalent to (3.14). This is identical to the equation for d for the boat model and the argument about the uniqueness of the solution carries over. □

3.4.2 The optimal setting

Let's now consider the optimal symmetric protocol. With similar reasoning, the expected latency of a player is

$$L_{OPT} = \sum_e \sum_{t=0}^{\infty} p_{e,t} \left(t + \ell_e(1) + (\ell_e(2) - \ell_e(1)) \sum_{r=-\ell_e(1)}^{\ell_e(1)} \left(1 - \tfrac{|r|}{\ell_e(1)}\right) p_{e,t+r} \right)$$

We seek probabilities with $\sum_{e,t} p_{e,t} = 1$ which minimize the above expression. Using a Lagrange multiplier and taking derivatives, we get that the minimum occurs when

$$\lambda = t + \ell_e(1) + 2(\ell_e(2) - \ell_e(1)) \sum_{r=-\ell_e(1)}^{\ell_e(1)} \left(1 - \tfrac{|r|}{\ell_e(1)}\right) p_{e,t+r}, \tag{3.15}$$

for some λ. The factor 2 in the last term comes from the convolution in the L_{OPT} expression.

We notice again the bicriteria property, that the optimal probabilities satisfy the Nash equilibrium condition of a different latency function.

Lemma 8. *The probabilities of the optimal solution for two players in the conveyor belt model of parallel links with latencies $\ell_e(k)$ is a Nash equilibrium for latencies $\ell'_e(k) = 2\ell_e(k) - \ell_e(1)$.*

Proof. By comparing Equations (3.11) and (3.15) that determine the Nash equilibria and the optimal solution, we see that the latencies must satisfy:

$$\ell'_e(1) = \ell_e(1) \qquad\qquad \ell'_e(2) - \ell'_e(1) = 2(\ell_e(2) - \ell_e(1)),$$

which can be expressed as in the lemma. □

3.4.3 The price of anarchy

Example. Consider one link with latencies $\ell_e(1) = 3$ and $\ell_e(2) = 19$. The probabilities $p_{e,t}$ at the Nash equilibrium, the costs $d_{e,t}$, and the optimal probabilities $p^*_{e,t}$ are

t	0	1	2	3	4	5	6	7	8	9	10	11	12
$p_{e,t}$	$25/48$	0	0	$1/3$	0	0	$7/48$	0	0	0	0	0	0
$d_{e,t}$	$34/3$	$34/3$	$34/3$	$34/3$	$34/3$	$34/3$	$34/3$	$104/9$	106/9	12	13	14	15
$p^*_{e,t}$	$31/80$	0	0	$47/160$	0	0	$1/5$	0	0	$17/160$	0	0	$1/80$

The cost at the Nash equilibrium is $d = 34/3 \approx 11.33$, while the optimal cost is $L_{OPT} = 1727/160 \approx 10.65$. The price of anarchy is approximately 1.05.

Since the conveyor belt Nash equilibrium and the optimal solution are very similar to the ones of the boat model, the analysis of the price of anarchy is similar. In particular, the expressions for the Nash equilibrium and the optimal solutions can be approximated well as the latencies $\ell_e(k)$ tend to infinity. For one link, the cost d of the Nash equilibrium is approximately $\sqrt{2\ell_e(1)(\ell_e(2) - \ell_e(1))}$ while the optimal cost is $\frac{4}{3}\sqrt{2\ell_e(1)(\ell_e(2) - \ell_e(1))}$, which shows that the price of anarchy tends to $3\sqrt{2}/4 \approx 1.06$, again. Since this is not sufficiently different than the boat model, we omit the details.

3.5 Discussion and Open Questions

Our results address some fundamental questions, but leave open important extensions.

For example, one can consider games with more general configurations, or adaptive strategies (based on the actions of the other players) that may even allow for preemption (abort the transmission and start over). In another variant of the game, players can wait before entering each edge of their own path.

Moreover we can consider other variants of the problem, such as letting only the past influence the delay in each link, or non atomic games.

Chapter 4

Game theoretic Modeling of the Worldwide Web

4.1 Introduction

The worldwide web has been the focus of an enormous amount of research in the last 15 years and several models have been proposed for it. These models aim at our understanding of the properties and evolution of the web, and assist us in designing more efficient web algorithms and applications (e.g. search engines). Recently, the exploitation of web's link structure by the search engines as well as the emergence of advertising links have given new incentives to link placement: strategic web page owners now explicitly attempt to boost their reputation and monetary revenue by careful selection of links, and Search Engine Optimization (SEO) has grown into a billion-dollar industry. Therefore Game Theory seems to provide the appropriate framework for studying the evolution of the web. Moreover, the impact of advertising links on the link structure of the web, and consequently on the relative importance of web pages, is unknown.

In this work we introduce a game-theoretic model for the worldwide web that captures the selfish nature of web page authors. In our model the page authors decide which advertising links to buy in order to maximize their revenue, which depends on the traffic their page attracts. We use Google PageRank as a measure of traffic. We study the extent to which these advertising links modify the PageRank and study whether it is possible to give the authors incentives to build a web of high total welfare in terms of its main application, the search for high quality pages.

4.1.1 Summary of Results and Techniques

We introduce and study a model for the web graph, in which selfish page owners aim at maximizing their PageRank and revenue by purchasing the appropriate incoming links to their page.

We introduce a model for the worldwide web, considering two different approaches for link pricing: fixed-prices and prices-per-click. Our proposed web game is not a potential game. Computing a Nash equilibrium of our game is NP-hard, so we compute an approximate best response with constant approximation ratio for the prices-per-click model.

4.1.2 Related Work

The first attempts to model the web graph coincide temporally with the development of successful web search algorithms which were based, partially, on the link structure of the web, with PageRank [109] and HITS [84] being the most well-known examples. The web search engines technology specified the notion of *importance* in the context of the web and motivated the research for understanding the web structure, aiming at our deeper understanding of the generative mechanisms driving the evolution of web, and the design of more efficient web algorithms. A description of the web graph structure is given in [39]. After mentioning the main properties of the web graph, we give a brief description of the measures of importance in the web, with emphasis on PageRank, as we will use it in analyzing our model. During the last fifteen years many models have been proposed for the web graph, which try to predict (some of) its structural properties. Classifying the models according to the deemed linking incentive, we get the following classes: *random graph models*, in which new nodes link to existing ones with high degree or PageRank; *economic models*, in which the nodes endorse existing ones that are regarded as good web search results; and *game theoretic models*, in which nodes explicitly try to maximize their own PageRank and/or revenue. We present shortly these models. Finally we present some closely related optimization problems that do not focus on game theoretic aspects.

Properties of the web graph. The structure of the web was a popular object of study about a decade ago. Many features of it have been examined thoroughly, including the main ones: the power-low distribution of its pages' degrees [39] and PageRank [26, 122], its diameter [6], the small world phenomenon [83], as well as many others, like the number of pages a site consists of [74], the absence of correlation between the age of a site and the number of its links [1], the site popularity [75], the self-similarity in web traffic [54, 55], the regularities in the surfing behavior of the users [76], and the heavy-tailed probability distributions of various features related to the web usage [56].

PageRank description. Surfers on the Internet use search engines to find pages satisfying their query. However there are typically hundreds or thousands of relevant

pages available on the web, and an unordered search results' list would not be practical. Modern search engines rank pages according to their importance, which they judge based on the link structure of the web. Google [37] uses *PageRank*, introduced by its founders in 1998 [109], as a measure of page importance. The link structure of n web pages is represented by the $n \times n$ matrix A with rows and columns corresponding to pages and

$$a_{ij} = \begin{cases} \frac{1}{d_i} & \text{if there is a link from } i \text{ to } j \text{ and } d_i > 0 \\ 0 & \text{if there is } \textit{not} \text{ a link from } i \text{ to } j \text{ and } d_i > 0 \\ \frac{1}{n} & \text{if } d_i = 0 \end{cases}$$

where d_i denotes the outdegree of page i. Assuming that a random surfer goes with some probability to an arbitrary web page with the uniform distribution, PageRank is defined as the stationary distribution of a Markov chain whose state space is the set of all web pages and the transition matrix is

$$\widehat{A} = cA + (1-c)\frac{1}{n}E,$$

where E is a matrix whose all entries are equal to 1 and $c \in (0,1)$ is the probability of not jumping to a random page (Google originally used $c = 0.85$). PageRank is the eigenvector π of the Google matrix $\widehat{A}$ (so $\pi^T \widehat{A} = \pi^T$) with $\pi^T \underline{1} = 1$, where $\underline{1}$ is a vector of ones. If a surfer follows a hyperlink of the current page with probability c and jumps to a random page with the rest probability $(1-c)$, then π_i is interpreted as the stationary probability that the surfer is at page i.

Random graph web models. The models proposed initially were mainly random graph (or stochastic) models that are online, since the number of nodes and edges varies with time, and produce graphs that possess most of the observed properties of the web, namely the power law degree distribution with exponent $\beta > 2$, the small world property (i.e., graph diameter much smaller than the order of the graph), and the existence of many dense bipartite subgraphs. All these models consider an evolving graph and determine the mechanism according to which each new node gets connected to (some of) the existing ones. We classify the random graph models based on the specific mechanism they employ, and distinguish between *preferential attachment* models, *geometric* models and *spatial preferential attachment* models, which lie in the intersection of the previous two classes. Other important classes of web models include the *off-line* models and the *copying* models, both of which overlap partially with the random graph models class.

Some of the first attempts to model the web growth were by Huberman et al. [74], Tadić et al. [121], Middleton et al. [99] and Kleinberg et al. [85]. The *preferential*

attachment mechanism has been widely used to model web growth. According to it, the new nodes are more likely to connect to existing ones with high degree. The first such models were the preferential attachment models of Albert and Barabási [23, 24] and Adamic et al. further improved on them [1], noticing that the age of a site is not correlated with the number of its links. The first rigorous attempt to design and analyze a preferential attachment web model was given in Bollobás et al. [33]. A set of other preferential attachment models were proposed by Aiello et al. [3]. These models are more complex than the LCD model, but yield graphs with power law exponent $\beta \in (2, \infty)$, dependent on the choice of some parameters, instead of the exponent $\beta = 3$ in the LCD model. Other important preferential attachment models include the model of Bollobás et al. [32] that uses preferential attachment in a way different than the previous ones; the model of Cooper and Frieze [52], which has a large number of parameters and thus becomes more complex; the models of Dorogovtsev et al. [62] and Drinea et al. [63] which introduce a variation of preferential attachment where each node is assigned a constant initial attractiveness and the probability that a new node is linked to an existing one is proportional to its in-degree plus this attractiveness; and a rigorous version of this model along the lines of the LCD model by Buckley et al. [41]. A preferential attachment model for random graphs in which each node exhibits some degree of fitness is introduced in [35]. Such a model could be used for the web graph since the web pages have some intrinsic value that is independent from the link structure and differentiates their attractiveness. Pandurangan et al. propose models that capture the power-law distribution of PageRank in the web [110]. In their basic model, which is similar to [63], attachment probabilities of new hyperlinks are based on the PageRanks of existing nodes. Another preferential attachment model based on PageRank, was recently proposed by Giammatteo et al. in [69]. Based on PageRank as well, Zhang et al. suggest a model that creates scale-free graphs in [124].

Economic web model. From another point of view, each web page is of specific utility (for the users) as a search result. It is natural therefore to assume that links are established in such a way that endorses the high utility search results. The page utility was taken into account in the economic model of Kouroupas et al [89, 90].

Game-theoretic web models. The decisions on the link structure of each web page are made locally, with each page owner trying to maximize the value and importance of her own page. Moreover, the increasing financial activity on the web (e.g., e-commerce, online advertising) has transformed web traffic to a potential source of revenue for the page owners. Hence the incentives in linking have become a bit more complex: apart from enriching the content of a page or assisting the page appear higher in the

search engines' results, links can also be purchased for advertising purposes. However, the money spent for advertising purposes shouldn't exceed the expected payoff from the traffic due to advertising links, and the link prices, as well as the link structure, depend on the rest of page owners. Therefore game theory appears as the proper framework for modeling the link establishment process, and we can think of the web as the equilibrium of some network creation game among its users.

The first network creation games were studied by Fabrikant et al. [65] and Anshelevich et al. [11]; these games produce internet-like networks, but are not proper for modeling the web graph. Tardos and Wexler [64] as well as Jackson [78] have surveyed the network formation games that have been proposed to model various types of strategic network formation (e.g., communication or social networks). During the last eight years, several aspects of linking in the web have been studied from a game theoretic point of view. In all these works, the link building process is modeled as a game among selfish web page authors who decide the link structure of their page in order to maximize their revenue from the web. The link structure may refer to either the *inner structure* of a web site, i.e., the links among the pages that constitute it, or the *external links* that connect it with other web sites, usually represented by a single page. External links may be advertising and/or reference (incoming to and outgoing from the node that establishes them, respectively), depending on the main purpose of their establishment.

The inner structure of a web site with strategic owner is studied in [77]. The authors consider two coalitional models for the transfer of PageRank among the site's pages and give the optimal linking structure of the site, that is computed in polynomial time. All other game theoretic models for the web concern the establishment of external links; either *reference only* [73, 43] or *reference and advertising* [80, 88]. There is also a line of research on the study of PageRank related games on *undirected graphs* [16, 17].

In the web reputation game of Hopcroft et al., [73], the players have to choose the set of reference links that maximize their PageRank (or *hitting time* [72] in another version of the game in the same paper). The model yields equilibrium graphs with a core. In [43] Chen et al. study the α-sensitive Nash equilibria of this game, that is the Nash equilibria that rely on the particular setting of the PageRank parameter α controlling the random jump probability.

In [80] Katona et al. study the formation of the commercial part of the web, i.e., the web pages designed with an eye to maximizing the revenue (monetary gain) of their owners. The players establish reference as well as advertising links, trying to maximize their revenue from traffic to their page. In the resulting equilibrium graphs, which have power low distributed degrees, the pages of high quality content earn money mainly

by selling content, while the low content pages' revenue comes from selling traffic to other pages. The drawback of this approach is that the players' payoffs at equilibrium are computed using the PageRank prior to the addition of the new nodes in the graph, instead of the PageRank after it. Building on the previous model, Kominers examines the strategic production of sticky content in commercial pages that generate revenues from both selling services and selling links [88]. However, since these approaches do not consider the effect of the new links on the pages that establish them, the question of what happens if we also consider this effect arises naturally. What is the game played among players that establish advertising links? Is choosing the appropriate advertising links harder than choosing reference links?

In another direction, recently, Avis et al. studied the Nash equilibria of PageRank related games on *undirected graphs* [16, 17] that can be used to model other networks, including some social networks. For instance, the 'friend' relationship on Facebook defines an undirected graph and PageRank can be used to measure the influence of a given user in it.

Various network creation games have been studied in terms of their efficiency, which is usually quantified by the price of anarchy [65, 11, 5, 60, 59, 61, 100, 53, 36, 25, 86].

PageRank related problems. There is also a large literature on local computations and optimization problems related to the web link structure, as well as on optimal linking strategies to maximize the PageRank of given nodes in directed graphs, that do not focus on game theoretic aspects.

In general, maximizing the PageRank via outlinks is easier than via inlinks, since the computations involve rank-1 updates of the hyperlink matrix instead of rank-k with $k \geq 1$. In [19] Avrachenkov et al. give a polynomial-time algorithm for maximizing the PageRank of a single node by selecting its outlinks. In particular, they study how the PageRank is affected by the addition of a set of new links emanating from a single page, and show that all out-going links of a page can be replaced by just one, carefully chosen link, without changing this page's PageRank. Kerchove et al. extend this result to maximizing the sum of the PageRanks of a given set of nodes (e.g. the pages that constitute a website) in [81]. Csáji et al. [57] give a polynomial-time algorithm for maximizing the PageRank of a single node with any given set of controllable links, using stochastic methods. The problems become harder when there are constraints among the links that can be selected or we have to select inlinks. In the above paper [57] it is shown that the problem becomes NP-hard if the set contains pairs of controllable links that are mutually exclusive. Moreover, Olsen has shown that maximizing the minimum PageRank in the given set of nodes is NP-hard if we are allowed to add just k new links

[104]; this is referred to as the *link building* problem.

Perhaps the closest in spirit to our web game classical optimization problem is the restricted version of the above where the node set is a single node and the k new links are inlinks to this node, which is proved by Olsen to be NP-hard [105]. In particular, it is shown there that given a graph $G = (V, E)$ and a node $t \in V$, computing a set of k new backlinks of t that maximize t's PageRank is $W[1]$-hard. In [107, 106] Olsen et al. give a constant factor approximation algorithm for this problem.

Another possible way to increase a node's PageRank is by colluding with other nodes. In [21] the authors study the impact of collusion in the PageRank of a set of nodes of the web graph. Another interesting issue is the behavior of PageRank as the web graph grows. The PageRank of growing networks is studied by Avrachenkov et al. in [18]. A recent study of the PageRank on evolving graphs is presented by Bahmani et al. in [22].

Since the web graph is huge, it is often useful to perform computations locally, i.e., examining only a small portion of the web graph near the nodes of interest. In [44] Chen et al. present methods for estimating PageRank values using only a small subgraph of the entire web. In [9] Andersen et al. compute the set of all nodes of the web graph that contribute to a specific node at least a constant fraction of its PageRank. This computation is performed locally as well. In [34] Borgs et al. give a sublinear randomized algorithm for computing the set of nodes with PageRank that exceeds some threshold value.

Figure 4.1 illustrates the aforementioned areas and the relations among them.

4.1.3 Organization

In the remaining chapter we introduce and study the game-theoretic aspects of link placement in the worldwide web. In Section 4.2 we show that a game in which web page authors establish hyperlinks aiming at the maximization of the PageRank of their page is not a potential game. We then present the models that have been introduced for the study of the establishment of reference links 4.3. In Section 4.4 we study the establishment of advertising links. We summarize our conclusions in Section 4.5 and give directions for future research.

4.2 Structural properties of link establishment

Consider a game played among the web page owners, in which each one aims at choosing the set of incoming links to her page that maximizes her page's PageRank. Assume

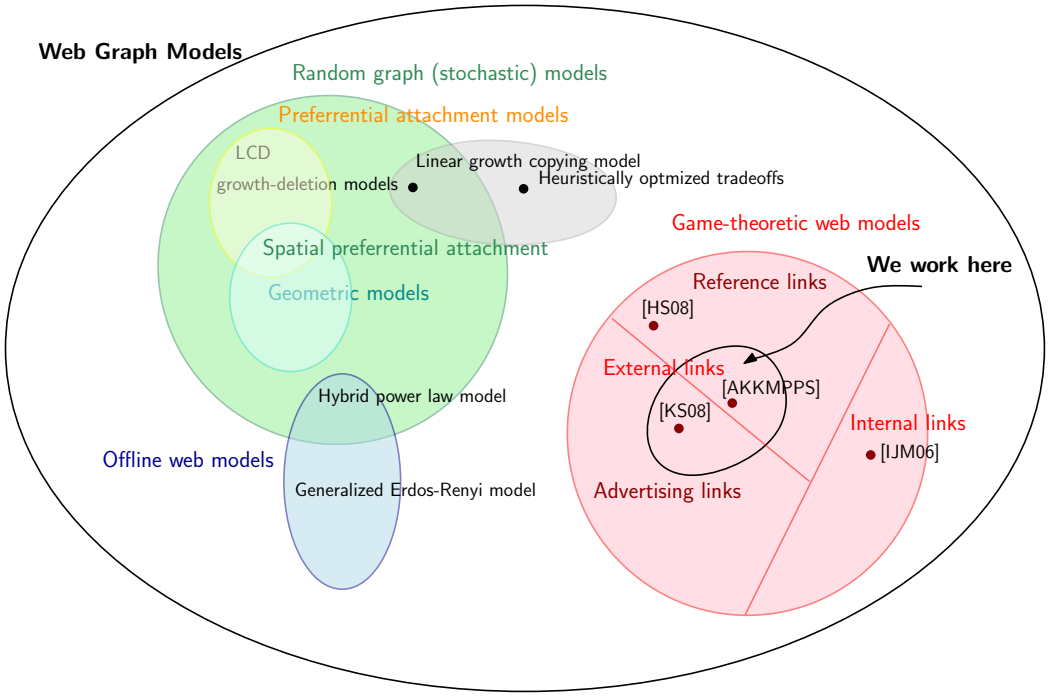


Figure 4.1: Related work

here that there is some restriction on the number of links that can be chosen, direct or indirect (through a cost for each link for example) – otherwise it would be in each player's interest to establish all possible incoming links, resulting to the complete graph. We call this game the *web game*. In this section we highlight the structural properties of such games.

The PageRank of each web page depends on the link structure of the web. In particular, as we saw in the definition of PageRank in Section 4.1, the PageRank each page receives through a specific incoming link depends on the total number of outgoing links of the page the link under consideration emanates from, i.e., the number of players that chose the same source node for a link to them. Therefore, among nodes of the same PageRank value, one would wonder whether it is preferable to get a link from the one that has the smallest number of outgoing links, or, more generally, whether this game is a congestion game.

Proposition 1. *The web game is not a congestion game for 3 or more players.*

Proof. The web game of 3 players is not a congestion game or equivalently, it does not admit an exact potential [101]. To verify this, consider the payoff of the players in the graphs with $E_0 = \{(1,2)\}$, $E_1 = \{(1,2),(3,1)\}$, $E_2 = \{(1,2),(2,3)\}$ and $E_3 = \{(1,2),(2,3),(3,1)\}$ shown in Figure 4.2, having PageRank vectors $\pi(E_0) =$

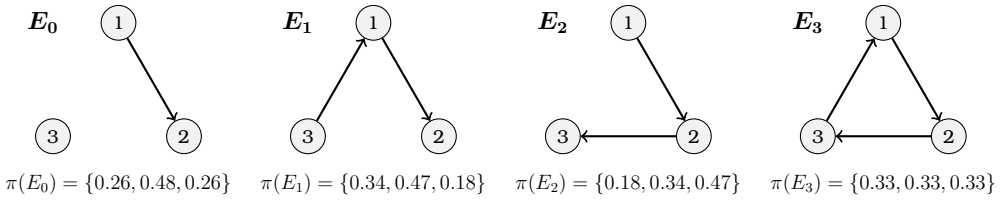


Figure 4.2: PageRank values on a simple graph.

$\{0.26, 0.48, 0.26\}$, $\pi(E_1) = \{0.34, 0.47, 0.18\}$, $\pi(E_2) = \{0.18, 0.34, 0.47\}$ and $\pi(E_3) = \{0.33, 0.33, 0.33\}$ respectively. If there exists an exact potential Φ, it must satisfy $\Phi(E) - \Phi(E') = P_i(E) - P_i(E')$, for any graphs G and G', with sets of edges E and E', which differ only on player i and for which $P(E)$ and $P(E')$ are the corresponding payoff vectors. In particular, this would imply

$$\pi_1(E_0) + p_{12} - \pi_1(E_1) + \pi_3(E_1) + p_{31} - \pi_3(E_3) = \pi_3(E_0) - \pi_3(E_2) + p_{23} + \pi_1(E_2) + p_{12} - \pi_1(E_3),$$

or equivalently

$$p_{31} - \pi_1(E1) = p_{23} - \pi_3(E_2).$$

Since this holds only for specific link prices p_{31} and p_{23} (that depend on the link structure of the graph), it follows that the web game is not a congestion game for 3 or more players. The case of 2 players is an exception as for $G = (V, E)$ the game has potential $\Phi(G) = \sum_{(i,j)\in E} p_{ij}$; it is not an interesting case however. □

This was expected, since the constraint $\pi^T \underline{1} = 1$ as well as the recursive nature of PageRank make the PageRank transferred from page i to j dependent not only on the number of pages that have an incoming link from i, but also on the PageRank of i itself and the PageRank of the rest pages (hence on the overall link structure).

The proposition holds for links with prices as well. The proof is almost the same for links with prices-per-click, and with slight modifications we can construct a similar counterexample for fixed link prices.

4.3 Establishing reference links

In [19] the authors study the effect of the establishment of a set of new links outgoing from a specific web page, to this page's PageRank. They prove that this establishment cannot affect the page's PageRank much, so it does not constitute a promising PageRank maximization strategy. In particular, a page, in order to maximize its PageRank through outlinks, has to build structures that are not meaningful in the context of the web. Moreover, they show that there is an optimal linking strategy in which a page can

replace all its outgoing links with a single, carefully selected link. However the complexity of these computations is not discussed. The Nash equilibria of games among players who establish outgoing links in order to gain reputation are studied in [73].

4.4 Establishing advertising links

4.4.1 The model

We aim at studying the web from a game theoretic point of view. In what follows we describe a web creation game. Web pages can gain reputation mainly from three sources: advertising links from search engines, Google adsense, and advertising links from other web pages. Here we study the third source only.

Consider the set of web page authors. Each one of them has a web page and wants to get the maximum possible payoff from it. Each page has content of some specific intrinsic value, and a number of links to/ from other pages. Representing each web site as a single page (e.g., the main page of the site) is a reasonable assumption; it is used by Google for example in PageRank computation, where they ignore all internal links of the site and assume that all incoming links to the site are directed to its main page and all its outgoing links emanate from it.

We model the web as a directed graph. The web pages and links correspond to the nodes and edges of the graph respectively. We assume that the graph consists of n nodes. Between two nodes i, j there is a directed edge (i, j), iff there is a link from page i to page j. The outgoing links on a page usually aim at enriching its content. In today's web, besides these (normal/ reference) links we also have *sponsored* (or paid/ advertising) links. Indeed, the page authors can purchase links: they can buy links (eg. from Google) or sell links (eg. through adwords). The Google search pages, for example, contain links of both kinds: the links that correspond to the search results (non sponsored) and the sponsored links that it sells to other pages.

We formulate the creation of the web graph as a game, in which:

- the set of players is the set of nodes of a graph,
- the available strategies of each player are *all possible sets of incoming links* she can purchase from other nodes
- the payoff of each player is the value of her page (its quality combined with the traffic it receives), minus the advertising cost, plus the income from selling outgoing links.

To be more precise, let $v_i \geq 0$ be the intrinsic quality of page i (the quality of its own content). The quality v_i expresses the revenue of page i per visitor. For any two pages i and j let us denote:

$$x_{ij} = \begin{cases} 1 & \text{if there is a link from page } i \text{ to } j \\ 0 & \text{otherwise} \end{cases}$$

Consider PageRank as a measure of the traffic a page receives. Moreover, assume that the cost of link (i, j) is equal to $p_i \ \forall j$.

We consider two pricing models for the advertising links: the *fixed-prices* and the *prices-per-click*. In the fixed-prices model, each node i offers advertising links at price p_i, so when link (i, j) is established, j pays p_i to i. In the prices-per-click model, i offers the links at price-per-click ρ_i, hence j pays ρ_i to i each time a visitor reaches j via i. The cost of (i, j) to j in this case is $p_i = d\frac{\pi_i}{d_i}\rho_i$.

The payoff of player i is:

$$P_i = v_i \cdot \pi_i + \text{cost of links sold by } i \text{ to others} - \text{cost of links bought by } i \Rightarrow$$

$$P_i = v_i \cdot \pi_i + \sum_{j=1}^{n} p_i x_{ij} - \sum_{k=1}^{n} p_k x_{ki}$$

and becomes (denoting by d_i the outdegree of page i)

$$P_i = v_i \cdot \pi_i + d_i \cdot p_i - \sum_{k=1}^{n} p_k x_{ki}$$

for the fixed-prices model and

$$P_i = v_i \cdot \pi_i + c\pi_i\rho_i - \sum_{k=1}^{n} c\frac{\pi_k}{d_k}\rho_k x_{ki}$$

for the prices-per-click model, so player i has to choose the set of advertising links she should establish in order to maximize it. Recall that the PageRank [109] of page i in a graph of n pages is defined as

$$\pi_i = \frac{1-c}{n} + c \sum_{k:k\to i} \frac{\pi_k}{d_k},$$

where c is the damping factor (usually set around 0.85). The total welfare of the game is the total payoff:

$$W = \sum_{i=1}^{n} P_i$$

To keep the model simple, we assume that nodes have no control over their outlinks, i.e., when a node wants to buy some link, the seller cannot refuse. (In real life some sellers may refuse to sell even if the price of the link is very high.)

We compute the PageRank based on the static web, but in fact the web is dynamic: there are pages that are created by search engines as the result sets of web searches. We'll try to overcome this fact considering the impact of the modern search engines in today's web structure and web pages.

In what follows, we assume that the v_i $\forall i$ as well as p (or ρ) are given as input to the problem.

4.4.2 On the hardness of computing best responses

We want to know whether a web graph $G = (V, E)$ is a Nash equilibrium of our web creation game. The obvious way to answer this question is to consider every player u, i.e. a node, and verify that the set $I_u(E) = \{r : (r, u) \in E\}$ of incoming edges to u is the best response for player u. But there are 2^{n-1} such sets and this obvious algorithm takes exponential time, so we need a better algorithm. One natural approach is the following: Player u checks only whether $I_u(E)$ remains her best response when she adds or removes an incoming edge to herself. That is, instead of checking all possible sets of incoming edges, she checks only that $I_u(E)$ is locally the best response, namely:

$$\text{for every node } r \in I_u(E) : v_u\ \pi_u(E) \geq v_u\ \pi_u(E \setminus \{(r, u)\}) + p$$

and

$$\text{for every node } r \notin I_u(E) : v_u\ \pi_u(E) \geq v_u\ \pi_u(E \cup \{(r, u)\}) - p$$

where p is the price of link (r, u). Is this test sufficient to verify that the set $I_u(E)$ is the best response?

This local check is sufficient when the following holds:

$$\pi_u(E \cup \{(r, u), (s, u)\}) + \pi_u(E) \leq \pi_u(E \cup \{(r, u)\}) + \pi_u(E \cup \{(s, u)\})$$

for any edges $(r, u), (s, u)$ that do not belong to E or equivalently, when PageRank is submodular. This is not true, as proposition 2 shows.

Proposition 2. *PageRank is not a submodular function.*

Proof. The PageRank function is submodular iff:

$$\pi_u(E \cup \{(r, u), (s, u)\}) - \pi_u(E \cup \{(r, u)\}) \leq \pi_u(E \cup \{(s, u)\}) - \pi_u(E).$$

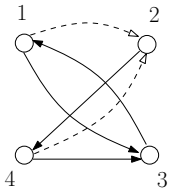


Figure 4.3: Initial graph and links under consideration.

for any nodes $u, r, s \in V : \{(r,u),(s,u)\} \notin E$.

Consider a graph $G = (V, E)$ with set of nodes $V = \{1,2,3,4\}$ and the edges $E = \{(1,3),(2,4),(3,1),(4,3)\}$ among them, shown in figure 4.3 (with normal edges). The above inequality does not hold for $u = 2, r = 1$ and $s = 4$ (dashed edges in figure 4.3). Indeed:

$$\pi_2(E \cup \{(1,2),(4,2)\}) - \pi_2(E \cup \{(1,2)\}) > \pi_2(E \cup \{(4,2)\}) - \pi_2(E)$$

$$(\Leftrightarrow 0.25 - 0.1719 > 0.0836 - 0.0375)$$

Intuitively, link $(1,2)$ creates the cycle $\{(2,4),(4,3),(3,1),(1,2)\}$, so since node 4 belongs to the cycle, adding link $(4,2)$ when the cycle exists increases π_2 more than adding it earlier. □

However, submodularity is not a necessary condition. What we precisely need to show is that whenever there is a set of incoming links to u that improve u's payoff, there is a single link among them that improves u's payoff as well. (Or, stated otherwise, if there is no single link that improves u's payoff, then no set of links can improve u's payoff either.) Below we show that this does not hold either.

Proposition 3. *The local check is not sufficient: A set of incoming links to a node may improve its payoff, even if none of those links improves the payoff on its own.*

Proof. Consider the graph G defined in proposition 2 (figure 4.3, normal links). Let $\Delta\pi_1 > 0, \Delta\pi_4 > 0$ and $\Delta\pi_{1,4} > 0$ denote the increase in PageRank of node 2 caused by the addition of links $(1,2)$ and $(4,2)$ in G and link $(4,2)$ in $G' = (V, E')$ with $E' = E \cup \{(1,2)\}$ respectively. Assume that these links, since they do not contribute in player 2's payoff, are priced $p_{12} = \Delta\pi_1 + \varepsilon$ and $p_{42} = \Delta\pi_4 + \varepsilon'$.

Applying lemma 10 twice, we notice that the PageRank of page 2 after the addition of the two links in G is of the form $\widetilde{\pi}_2 = \pi_2 + \Delta\pi_1 + \Delta\pi_{1,4}$, so the difference in payoff is $\Delta\pi_1 + \Delta\pi_{1,4} - p_{12} - p_{42} = \Delta\pi_1 + \Delta\pi_{1,4} - \Delta\pi_1 - \varepsilon - \Delta\pi_4 - \varepsilon' = \Delta\pi_{1,4} - \Delta\pi_4 - \varepsilon - \varepsilon'$. In proposition 2 we saw that $\Delta\pi_{1,4} > \Delta\pi_4$ ($\Delta\pi_{1,4} = 0.25 - 0.1719$ and $\Delta\pi_4 = 0.0836 - 0.0375$) so for sufficiently small ε and ε' the above difference is positive and the set of links $\{(1,2),(4,2)\}$ increases player 2's payoff. □

Finding the best response for each player given the strategies of the rest players seems to be computationally hard. As the following theorem states, it is indeed NP-hard, since the LINK BUILDING problem, which is known to be NP-hard (in fact it has no FPTAS unless $P = NP$) [105], reduces to it.

Definition 2. LINK BUILDING problem:
Given a triple (G, u, k) where $G = (V, E)$ is a directed graph, $u \in V$ and $k \in \mathbf{Z}^+$, find a set $S \subseteq V \setminus \{u\}$ with $|S| = k$ maximizing $\widetilde{\pi}_u(S \times \{u\})$.

Theorem 6. *Given a directed graph $G = (V, E)$, a node $u \in V$ and the price p_i of the links emanating from node i for all $i \in V$, the problem of computing the set of new backlinks of u that maximize $\pi_u - \sum_{j:j\to u} p_j$ is NP-hard.*

Proof. We prove that the above problem (P) is NP-hard, reducing the *Link Building* problem to it. In particular, we show how to solve an instance of the *Link Building* problem in polynomial time if we have a polynomial algorithm $\mathcal{A}_P$ to problem P.

Consider an instance of P in which all prices are equal, i.e., $p_i = p \ \forall i \in V$. We execute $\mathcal{A}_P$ for this instance. If u has k new incoming links in the solution, we have solved the *Link Building* problem. Otherwise we increase/reduce p and execute $\mathcal{A}_P$ again for the new instance. It remains to show that we can reach k in a *polynomial* number of tries.

Consider the best responses s and s' of u in two instances of our game with link prices $p_s, p_{s'}$ respectively, and let b_s and $b_{s'}$ be the corresponding numbers of new backlinks to u.

Lemma 9. *If $b_{s'} > b_s$ then $p_{s'} < p_s$. Equivalently, if $b_{s'} < b_s$ then $p_{s'} > p_s$.*

(The proof of lemma 9 is presented after the current proof.)

Therefore in order to increase/reduce the number of new backlinks to u in u's best response, we have to reduce/increase the link price p appropriately. Assume that we executed $\mathcal{A}_P$ for some instance of P and got a solution with $k_0 > k$ new incoming links. We want to reduce the new links in the solution to k so we increase the link price repeatedly, each time achieving a solution with 1 new link less than the previous one. The price we will reach after $k_0 - k$ steps is one of the prices for which the best response consists of exactly k new links. It remains to specify the increase in each step.

Wlog we assume that u has no incoming links before she starts playing. The payoff of u playing strategy s, denoting with $\pi_u(s)$ the corresponding PageRank of u, is

$$P_u(s) = \pi_u(s) - \sum_{v:v\to u} p_v = \pi_u(s) - pk_0$$

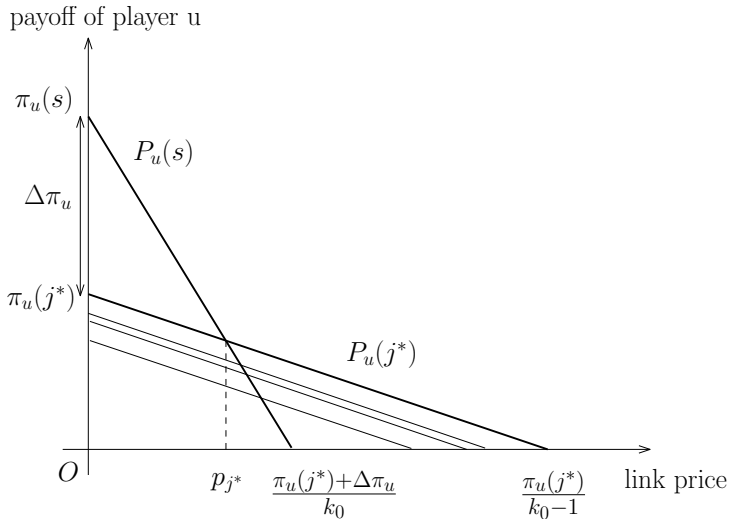


Figure 4.4: Payoff of player u as a function of the link price.

which is linear in p. Strategy s is a best response as long as $P_u(s) > P_u(j^*)$ where j^* is some strategy with $k_0 - 1$ new links, in fact $j* = \arg\max_j\{P_u(j)\} = \arg\max_j\{\pi_u(j) - p_j(k_0 - 1)\}$. Both these payoffs, as well as example payoffs of other strategies of $k_0 - 1$ new links (i.e., the lines that are parallel to $P_u(j^*)$ and below it), are demonstrated as functions of the link price in figure 4.4. $P_u(j^*)$ is decreasing linearly in p_j as well, and since its maximum value ($\pi_u(j^*)$) is lower than $\pi_u(s)$ and its gradient is less steep, it follows that as the link price increases, after some point $P_u(j^*)$ exceeds $P_u(s)$. This happens at the price p_{j^*} for which $P_u(s) = P_u(j^*)$, i.e., for $p_{j^*} = \pi_u(s) - \pi_u(j^*)$ and from lemma 10 we get that $p_{j^*} = \pi_v \frac{cz_{uu}-z_{vu}}{v_0+cz_{uv}-z_{vv}}$ where v is the node, among the ones included in the set of backlinks $j*$, that yields the minimum decrease in u's PageRank if removed from j^*, so $v = \arg\min_{w\in s} \pi_w \frac{cz_{uu}-z_{wu}}{w_0+cz_{uw}-z_{ww}}$.

Therefore if we have a polynomial algorithm $\mathcal{A}_P$ that computes the best response in our game, we can use it to solve the *Link Building* problem efficiently: If the solution does not happen to consist of k new links, we increase/reduce the link price repeatedly as specified above, reaching in each step solution of one link less/more than the previous one respectively, until we reach k links. This computation takes polynomial number of steps ($|k - k_0|$). However, *Link Building* is NP-hard so P is NP-hard as well. □

Proof. (Lemma 9) If p_s was lower than $p_{s'}$, we could increase the payoff of t by switching from s to s' and get higher PageRank (same as in s') while paying less than in s', so s' wouldn't be a best response for $p_{s'}$. Based on this idea, we prove the lemma by contradiction.

Let us denote with $\pi_t(j)$ the PageRank of node t when playing strategy j. The strategy s' is the best response of t for price $p_{s'}$, hence $\pi_t(s') - b_{s'}p_{s'} \geq \pi_t(j) - b_j p_{s'}\ \forall j \neq s'$ and

so

$$\pi_t(s') - b_{s'}p_{s'} \geq \pi_t(s) - b_s p_{s'}. \tag{4.1}$$

Similarly, strategy s is the best response of t for price p_s, hence $\pi_t(s) - b_s p_s \geq \pi_t(j) - b_j p_s \; \forall j \neq s$ and so

$$\pi_t(s) - b_s p_s \geq \pi_t(s') - b_{s'} p_s. \tag{4.2}$$

Assume that $p_{s'} > p_s$, for example $p_{s'} = p_s + \Delta p$ with $\Delta p > 0$. We sum (4.1) and (4.2) and get $-b_{s'}p_{s'} - b_s p_s \geq -b_s p_{s'} - b_{s'} p_s \Rightarrow$

$$(b_{s'} - b_s)p_s \geq (b_{s'} - b_s)p_{s'} = (b_{s'} - b_s)p_s + (b_{s'} - b_s)\Delta p \Rightarrow (b_{s'} - b_s)\Delta p \leq 0 \Rightarrow b_{s'} \leq b_s.$$

The last inequality contradicts the assumption in the statement of the lemma, so $p_{s'} > p_s$ does not hold.

We derive the second statement of the lemma by reversing the inequalities in the above paragraph. □

It follows that verifying a Nash equilibrium is NP-hard as well, since we have to verify that each player plays his best response given the strategies of the rest players.

Corollary 1. *Verifying a Nash equilibrium in our web game is NP-hard.*

Finding *any* set of backlinks that yields larger payoff than the current one, even if it does not maximize it, can not be done efficiently either, since as we saw in proposition 3 we may have to check all possible sets.

Moreover, even characterizing the equilibria is a challenging task. The necessary and sufficient conditions for equilibrium existence include rank-k updates of the Google matrix, which are only expressed by involved formulas (see rank-2 updates in lemma **??** for example) that make it impossible for us to derive readable conditions. We note here that in the model of Katona et al. [80] where they compute the equilibria of a very similar model, they assume that the PageRank of the source of the link is not affected by the establishment of the link, which is not correct. In particular, in their model player j computes the payoff she will get *after* the establishment of link (i, j), that depends on π_j, using the value of π_j *before* it; however π_j may decrease with the establishment of (i, j).

The computation of best responses may be easy in *special cases* of graphs. For instance, consider a game in which all intrinsic values are equal and all link prices (fixed or per-click) are equal, say v and p respectively. Then all players are symmetric and we have to analyze just one of them.

Proposition 4. *The web game on graphs with nodes of equal intrinsic values and edges of equal prices possesses at least one pure Nash equilibrium.*

Proof. (Sketch): The players are symmetric so at equilibrium they all have the same link structure. A configuration in which no player can increase her payoff by buying any set of advertising links is a pure Nash equilibrium. Such an equilibrium always exists. The density of equilibrium graphs depends on the ratio $\alpha = \frac{p}{v}$. For small values of α the only equilibrium is the complete graph, while for large values of α only the empty graph is equilibrium. The rest regular graphs are equilibria for the intermediate values of α, with decreasing node degree as α increases. □

4.4.3 Approximate best responses

In [107, 106] the authors propose a constant factor approximation algorithm for the link building problem. We can employ it to compute an *approximate best response* in our game. (The authors also compare this approximation algorithm to the naive algorithm where we choose backlinks from nodes with high PageRank values compared to their outdegree and show that, on certain graphs, the latter performs much worse than the former.)

We consider a game of the prices-per-click model. In this case player u's best response is the strategy s that maximizes

$$P_u(s) = v_u \pi_u(s) - \sum_{k:k\to u} c\frac{\pi_k(s)}{d_k}\rho_k = \frac{1-c}{n}v_u + c\sum_{k:k\to u}(v_u - \rho_k)\frac{\pi_k(s)}{d_k},$$

which, since $\pi_u = \frac{1-c}{n} z_{uu}\big(1 + \sum_{j=1,j\neq u}^{n} q_{ju}\big)$, can be written as (proof!):

$$P_u = \frac{1-c}{n} z_{uu}\left(1 + \frac{1}{v_u}\sum_{j=1,j\neq u}^{n}(v_u - \rho_j)q_{ju}\right).$$

We can use a greedy algorithm to compute an approximate best response. Algorithm 4.5 runs in polynomial time (it requires n^2 steps in worst case).

Proposition 5. *Let $P_u'^G$ denote the payoff obtained by the solution of algorithm 4.5 and $P_u'^o$ the payoff yielded by the best response. We have*

$$P_u'^G \geq P_u'^o(1-c^2)\left(1 - \frac{1}{e}\right).$$

Proof. We have $\frac{P_u}{z_{uu}} = \frac{1-c}{n}\left(1 + \frac{1}{v_u}\sum_{j=1,j\neq u}^{n}(v_u - \rho_j)q_{ju}\right)$. Algorithm 4.5 considers the links whose price per click is less than v_u and greedily adds backlinks to u attempting to maximize the probability of reaching u before absorption, with the minimum cost. q_{ju} is a nonnegative, submodular function of the set of backlinks of u, that is nondecreasing

procedure APPROXBESTRESPONSEPRICEPERCLICK(G,u,ρ)
  $Smax := \emptyset$
  maxpayoff $:= 0$
  $S := \emptyset$
  $V' := \{i \in V : (i,u) \notin E \ \wedge \ \rho_i < v_u\}$
  for $k = 1$ **to** $|V'|$ **do**
    $r := \arg\max_j \frac{P_u}{z_{uu}}$ in $G = (V, E \cup \{(r,u)\})$ where $r \in V'$
    $S := S \cup \{r\}$
    $E := E \cup \{(r,u)\}$
    if $P_u >$ maxpayoff **then**
      $Smax := S$
      maxpayoff $:= P_u$
  return $Smax$

Figure 4.5: approxBestResponsePricePerClick(G,u,ρ): Pseudocode for finding an approximate best response in the price-per-click game

after adding any backlink. These properties are preserved if we multiply each q_{ju} by the utility $(v_u - \rho_j)$ that link (j,u) incurs to u (i.e., increase in PageRank minus link cost). Indeed, it is nonnegative and nondecreasing since $v_u - \rho_j > 0$ for all links (j,u) considered, and submodular because the effect of $v_u - \rho_j$ on the q_{ju} is independent of the graph structure at the time when link (j,u) is added. Hence $\frac{P_u}{z_{uu}}$ is a nondecreasing and submodular function as well, as the sum of nondecreasing and submodular terms.

Let $P_u'^G$ and $z_{uu}'^G$ denote the values obtained by algorithm 4.5, $P_u'^o$ the optimal value and $z_{uu}'^o$ the value of z_{uu} corresponding to $P_u'^o$. According to [**?**] we have that

$$\frac{P_u'^G}{z_{uu}^G} \geq \frac{P_u'^o}{z_{uu}^o}\left(1 - \frac{1}{e}\right),$$

where $e \approx 2.71828$. Since $z_{uu} = \frac{1}{1-q_{uu}}$ (from [19]) and $q_{uu} \in [0, c^2]$, it is $z_{uu} \in [1, \frac{1}{1-c^2}]$, so $\frac{z_{uu}^o}{z_{uu}^G}$ is maximized when $z_{uu}^o = \frac{1}{1-c^2}$ and $z_{uu}^G = 1$. Therefore $\frac{P_u'^o}{P_u'^G} \leq \frac{1}{1-c^2}\frac{e}{e-1}$. For $c = 0.85$ this gives an upper bound of $\frac{P_u'^o}{P_u'^G}$ of approximately 5.7. Note that if z_{uu} cannot improve much with the addition of new backlinks, i.e., it is already close to its optimal value, z_{uu}^G is close to $\frac{1}{1-c^2}$ and the upper bound of $\frac{P_u'^o}{P_u'^G}$ drops to approximately $\frac{e}{e-1} \approx 1.58$.

□

4.4.4 Establishing a single advertising link

In this section we consider (pure) Nash equilibria of n players on directed web graphs. A pure strategy for a player is to select a subset of new incoming links to be established. Intuitively, player u establishes inlinks with the goal of maximizing the expected number of times a random walk on the graph visits u, i.e., inlinks from nodes that are visited

often, at a relatively low cost.

Definition 3 (Nash equilibrium graph)**.** A web graph is at Nash equilibrium iff no node has an incentive to change its set of incoming links unilaterally.

The game is finite, so according to Nash [102] it possesses at least one mixed Nash equilibrium. We are interested in pure equilibria.

We first study the effect, on PageRank, of adding or removing incoming links to/from a specific node. Then we deal with the Nash equilibria of the two models.

We consider the establishment of a single advertising link for each node as the choice of the best such link can be done in polynomial time. We call this game the *single-link* game.

4.4.4.1 The effect of adding/removing incoming links (backlinks)

From the definition of PageRank we get that $\pi = \frac{1-c}{n}[I - cA]^{-T}\underline{1}$ (since $\pi^T(cA + (1-c)\frac{1}{n}\underline{1}\underline{1}^T) = \pi^T I \Leftrightarrow \pi^T(I - cA) = \frac{1-c}{n}\underline{1}^T \Leftrightarrow \pi^T = \frac{1-c}{n}\underline{1}^T[I - cA]^{-1}$). We now compute the change in a page's PageRank caused by the addition or removal of an incoming link.

Lemma 10. *Consider a graph $G = (V, E)$ with $r, u \in V$. Let π_u denote the PageRank of page u, and d_r the outdegree of r.*

- *If $(r, u) \notin E$, the addition of link (r, u) changes u's PageRank to*

$$\widetilde{\pi}_u = \pi_u + \pi_r \frac{cz_{uu} - z_{ru}}{d_r - cz_{ur} + z_{rr}}.$$

- *If $(r, u) \in E$, the removal of link (r, u) changes u's PageRank to*

$$\widetilde{\pi}_u = \pi_u - \pi_r \frac{cz_{uu} - z_{ru}}{d_r + cz_{ur} - z_{rr}}.$$

Proof. The addition of link (r, u) can be regarded as a rank one update of the hyperlink matrix A, namely $\widetilde{A} = A + e_r b_+^T$ with $b_+^T = \frac{1}{d_r+1}e_u^T - \frac{1}{d_r+1}a_r^T$, where a_r^T is the r-th row of matrix A. Let $[I - cA]^{-1} = Z$.

Applying the Sherman-Morrison-Woodbury updating formula to $[I - c\widetilde{A}]^{-1}$ we get

$$[I - c\widetilde{A}]^{-1} = [I - cA]^{-1} + c\frac{[I - cA]^{-1}e_r b_+^T[I - cA]^{-1}}{1 - cb_+^T[I - cA]^{-1}e_r}.$$

So

$$\widetilde{Z}^T = Z^T + c\frac{Z^T b_+ e_r^T Z^T}{1 - cb_+^T Z e_r}.$$

Postmultiplying this equation by $\frac{1-c}{n}\underline{1}$ we get

$$\widetilde{\pi} = \pi + c\frac{Z^T b_+ e_r^T}{1 - cb_+^T Z e_r}\pi$$

and consequently

$$\widetilde{\pi}_u = \pi_u + c\frac{e_u^T Z^T b_+ e_r^T}{1 - cb_+^T Z e_r}\pi = \pi_u + c\frac{\frac{1}{d_r+1} e_u^T Z^T (e_u - a_r) e_r^T}{1 - c\frac{1}{d_r+1}(e_u^T - a_r^T) Z e_r}\pi$$

$$= \pi_u + c\frac{e_u^T Z^T (e_u - a_r)}{d_r + 1 - c(e_u^T - a_r^T) Z e_r}\pi_r = \pi_u + c\frac{e_u^T Z^T e_u - e_u^T Z^T a_r}{d_r + 1 - c(e_u^T Z e_r - a_r^T Z e_r)}\pi_r$$

$$= \pi_u + \frac{cz_{uu} - ce_u^T Z^T a_r}{d_r + 1 - cz_{ur} + ca_r^T Z e_r}\pi_r$$

We evaluate $ce_u^T Z^T a_r$ and $ca_r^T Z e_r$. Since $[I - cA]Z = I$ we have

$$\left\{ \begin{array}{l} cAZ = Z - I \\ cZ^T A^T = Z^T - I \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} ce_i^T AZ e_r = e_i^T (Z - I) e_r \\ ce_i^T Z^T A^T e_r = e_i^T (Z^T - I) e_r \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} ca_i^T Z e_r = z_{ir} - e_i^T e_r \\ ce_i^T Z^T a_r = z_{ri} - e_i^T e_r \end{array} \right.$$

so $ca_r^T Z e_r = z_{rr} - 1$ and for $u \neq r:\ ce_u^T Z^T a_r = z_{ru}$ and therefore

$$\widetilde{\pi}_u = \pi_u + \frac{cz_{uu} - z_{ru}}{d_r + 1 - cz_{ur} + z_{rr} - 1}\pi_r.$$

Similarly, the removal of link (r, u) is a rank one update of the hyperlink matrix $\widetilde{A} = A - e_r b_-^T$, with $b_-^T = \frac{1}{d_r - 1}(e_u^T - a_r^T)$. We get that

$$\widetilde{Z}^T = Z^T - c\frac{Z^T b_- e_r^T Z^T}{1 + cb_-^T Z e_r}.$$

and the second formula of the lemma follows. □

To compute rank-k updates we can employ Sankowski's algorithm [117].

The incoming links always contribute positively to the receiving page's PageRank.

Lemma 11. *A single link that is added to a graph always increases the receiving page's PageRank.*

(Note: If the source node establishes more outgoing links, (especially) to 'irrelevant' to the receiver above nodes, the PageRank may reduce as well. However we are not interested in this case, since those links are part of other players' strategies, who are assumed not to deviate.)

Consequently, removing an incoming link from a page always reduces its PageRank. Moreover, if more than one new links with the same target are established, they always improve the receiving page's PageRank.

Lemma 12. *Any set of incoming links to a page improve its PageRank.*

Proof. The addition of a set of k links to the graph, all pointing to a specific page u, is equivalent to a successive addition of these links, one by one. By lemma 11 each link contributes positively in u's PageRank, regardless the structure of the graph, hence the whole set contributes positively as well. □

The payoff of a player, however, increases only if the total cost of the new incoming links is strictly lower than the increase they yield in her page's PageRank.

Consider a restricted version of the initial game, in which each player is allowed to purchase *only one* advertising link. We 're interested in characterizing the Nash equilibria of this game, i.e. the comparison of the Nash equilibrium graph with the optimal structure.

Theorem 7 (Nash equilibria of the restricted game)**.** *A graph is a Nash equilibrium for the single-link game iff for any pair of nodes (u, r) such that the edge (r, u) does not exist in the graph, it is $p_r > v_u \pi_r \frac{cz_{uu} - z_{ru}}{d_r - cz_{ur} + z_{rr}}$.*

Proof. It must be $p_r > v_u \pi_r \frac{cz_{uu} - z_{ru}}{d_r - cz_{ur} + z_{rr}}$, otherwise according to lemma 10, node u could purchase an incoming link from r and increase its payoff. □

4.4.5 The Optimal structure

The optimal structure of the graph is the one that maximizes the total welfare. Search engines use measures of importance of the pages, like PageRank, in order to approximate their quality. In terms of our model, for each page i they use π_i (which they compute based on the link structure) as an estimation of v_i (which can not be determined since it depends on the content of i and numerous other factors). Hence a web graph is 'good' if π_i is proportional to v_i for any page i, and in the optimal web graph it is $\frac{\pi_i}{v_i} = \alpha$ for every node i.

4.5 Discussion and Open Questions

The establishment of appropriate reference and advertising hyperlinks is a common practice of web page authors in order to increase the reputation of their pages. However,

computing the set of advertising hyperlinks that maximize a page's reputation is an NP-hard problem. We suspect that computing a fixed plurality set of reference hyperlinks that maximize a page's reputation is not easier, but for now this problem remains open.

Moreover, we propose an algorithm that, given the prices-per-click of the links, computes an approximate best response for a page author. Is there an approximate algorithm for the case of fixed price links? Moreover, can we model the link establishment as a congestion game, in order to find some approximate best response and Nash equilibrium?

Part III

Compact Representation of Routing and Information Networks

Chapter 5

Compact Network Representation

5.1 Introduction

Real-world systems and phenomena that involve interactions among various entities are being modelled using graphs for decades now. The recent explosive growth of large-scale systems that are traditionally modelled as graphs, the worldwide web and social networks being typical examples, has intensified the need for compact, yet efficient, representations of graphs. In particular, we need compressed graph representations that allow mining without decompressing the graph. In this way, algorithms and applications with tasks that correspond to graph mining problems, can take advantage of such representations to boost their performance, as they can run in main memory over much larger graphs using their compressed representations instead of the plain ones. For example, serving adjacency queries or maintaining and querying low-cost snapshots for archival purposes are common operations in such critical applications, and can benefit from the use of in-memory representations of graphs.

The graphs we are interested in representing share some common features. First, they represent huge networks extending to millions of nodes, but the degrees (in/out-degrees) of the latter are power law distributed [45, 40], rendering the graphs to be rather sparse [58]. Moreover, the graphs exhibit the *locality of reference* property: nodes tend to have successors that are 'close' to them in a sense that depends on the context and the nature of the network. For instance, web pages often contain links to pages of the same web site or domain, and people in social networks are often friends with individuals from the same neighbourhood, university, or work. Furthermore, these graphs exhibit the *copy* property (or *similarity* property), which denotes that nodes occurring close to each other tend to have many common successors.

These properties induce various types of redundancy in the graphs' representations, and are taken into account when designing compression methods. The state-of-the-art approach to the compact representation of graphs is the method of Boldi and Vigna [29], further improved using a reordering of the graph [30] before compressing it. Several

other approaches have been proposed, but they are slower, i.e., they improve the results of [29] only in terms of compression ratios and not in terms of access times of the graph's elements, they are efficient for small graphs only, or they are methods solely based on some usually computationally expensive reordering of the input graph. We note here that reordering a given graph results in an isomorphic graph, in which redundancy can be (hopefully) exploited by the algorithm more effectively. For example, in [31, 30] the authors introduce reorderings for which the Boldi and Vigna method yields an increased compression of web and social network graphs, when compared with the compression obtained using the graphs in their initial form. Hence, the reorderings can favour any compression algorithm that takes the aforementioned properties into account.

The web and social graphs may share the above properties, but feature a substantial difference in the way they are represented: while it is easy to order the nodes of a web graph in a meaningful way which favours its compression, there is no such obvious ordering for general networks, including social ones. As it is noted in [45], there exists some, yet unexplained, topological difference between social networks and web graphs that results in a less effective compression of the former, i.e., a larger compression ratio.

5.1.1 Summary of Results and Techniques

We concentrate on the compression of web, social network and other similar graphs by exploiting the locality property. After observing that the above types of graphs, as well as most graphs created by human activity, demonstrate the locality property, i.e., they can be represented by adjacency matrices with high concentration of edges around the main diagonal of the matrix, we exploited this fact to improve the compression of such graphs.

Since the highest compression ratios are achieved by the state-of-the-art algorithm of Boldi et al., namely, the compression framework of [29] (denoted as *BV*) after applying the *Layered Label Propagation* (LLP) algorithm [30] on the input graphs, we decided to build on it, making the following contributions:

- we improve *BV* by exploiting the locality of reference property observed in these kinds of graphs in a different way than in [29] and, thus, go beyond the state-of-the-art in graph compression

- we evaluate experimentally our algorithm and show that it achieves a better compression ratio than *BV*, while allowing the retrieval of elements of the graph faster than *BV*.

Of course, our results further improve if we apply our algorithm on a reordered version of our input graph, using for example the reordering algorithm of [30].

5.1.2 Related Work

The need for compact representation of graphs emerged with the explosion of the size of the worldwide web, so the first such attempts focused on compressing web graphs. In the last dozen of years graph compression has turned into a very active research area and many algorithms have been proposed, some of them designed for more general graphs like the social network ones. Most algorithms in this direction try to offer a good space/time trade-off.

The structures traditionally used for the representation of graphs are the adjacency list and the adjacency matrix. The former is preferred for sparse graphs, i.e., graphs whose number of edges is $O(n)$, where n denotes the number of the graph's nodes, while the latter is used for dense graphs, i.e., graphs with $\Theta(n^2)$ edges. The locality of reference property, as well as the node similarity property, have been observed in most of the graphs we are interested in, and are often met in graphs that represent networks created by human activity. The central idea in graph compression algorithms is that they try to diminish the inner redundancy in the representation using the above structures, by exploiting the aforementioned properties.

The graph compression algorithms that have been proposed so far can be classified in the following three main categories: (i) algorithms for compressing web graphs, (ii) algorithms for compressing (also) more general graphs (mostly social network graphs), and (iii) algorithms that include or employ reordering of the graph in order to favour higher degree of compression. It is also very often the case that specific web graph compression algorithms were later enriched with new techniques in order to be able to compress social graphs as well.

In [112] the authors take into account the locality of reference and the copy properties for the case of the web and initiate research on web graph compression by maintaining compressed forms of the graph's adjacency lists. The highest compression ratios are achieved by the method of Boldi and Vigna [29], combined with a reordering using label propagation [30]. The WebGraph compression method introduced in [29] is indeed the most successful member of a family of approaches [111, 40, 2, 120] for compressing *web* graphs based on the statistical properties described in the introduction. In [29] Boldi and Vigna exploit the similarity of adjacency lists and the locality of reference of nearby pages using URL ordering for nodes. We present the techniques of the WebGraph framework briefly in Section 5.3.1.1.

In another line of work, Brisaboa et al. [38] propose a compact representation of the

adjacency matrix that represents the graph. They partition the matrix in boxes and store each box in a way that allows quick access to it. In particular, they use a k^2-ary tree that records at each level which children contain at least one edge. The most important feature of this work is that it allows both forward and backward navigation of the graph. However, experiments show that even for the smallest possible value of k, viz., 2, which results in 4-bit sized leaves, the total compressed size wasted on leaves of the tree alone, is significantly greater than that achieved by other methods for graphs of greater size than the ones tested. The approach we propose in this work is to some extent similar to [38], in the sense that we represent parts of the adjacency matrix of a given graph. The difference with our approach is that we represent only some *dense* parts of the graph, those that are close to the main diagonal, and that we do not introduce extra overhead by using trees as indices.

Asano et al. [15] reorganize the adjacency matrix of the graph to bring the inter-host links close to the intra-host ones, and incorporate six different kinds of patterns to cover it. The compression ratio reported is impressive, but the additional cost imposed for the matching of the original indices of the inter-host links with the new local indices used is not considered in the presented results. This approach is similar to our proposed method, with the difference that in our method no overhead for lookup tables is introduced.

Claude et al. in [51] use a different compression scheme for web graphs that does not achieve better compression ratios than [29], but allows for faster navigation on the graph. However, their approach does not scale up due to the large amount of memory and long time required during the computationally expensive compression phase. Later on, Claude et al. combined this algorithm with the techniques of [38], partitioning the input graph and applying a separate technique to each part (i.e., applying [51] on one part and [38] on the other), to compress web as well as social network graphs [50]. A similar partitioning is applied in our approach as well, but the methods used to compress the subgraphs are entirely different.

In [45] Chierichetti et al. view the problem of graph compression from a theoretical point of view and study the extent to which a large social network can be compressed. They show that the compressibility of social networks is very different than that of web graphs. Their proposed method, however, is a compression scheme rather than a compressed data structure, as noted in [30], i.e., it aims solely at minimizing the size of the compressed graph (bits/edge) instead of providing fast access to each edge.

The locality of reference property of a graph reflects on its adjacency matrix in the following way: using a proper ordering of the nodes' labels, i.e., an ordering in which labels of densely connected nodes are close to each other, many edges fall close to the

main diagonal of the adjacency matrix. Such orderings are preferred in practice, but finding the ordering that minimizes the distance of the edges from the main diagonal is NP-hard [119]. Intuitively, if we have some good clustering of the graph, based solely on the link structure, and assign consecutive labels to the nodes in each cluster, the lexicographic ordering of the labels is rather good in the above sense.

Such an extrinsic ordering appears naturally in the case of worldwide web. Web graph representations assume that each URL corresponds to some identifier. Moreover, it is assumed that URLs are alphabetically sorted [27], and this naturally puts together the pages of the same domain. As a result, the locality of reference translates into closeness of page identifiers. However, extrinsic orderings are not obvious for all graphs, so for social or bibliographic citation graphs, finding a good ordering is a challenging issue. In [31] Boldi et al. test some known orderings of the nodes and propose some new ones, and study their effect on the compression of web and social graphs. They show that using these orderings for the input non-web social graphs, the WebGraph framework [29] yields results that are very close to the results of [45]. In [30] the authors introduce a reordering algorithm called *Layered Label Propagation* (LLP), and employ it to compress social networks. This algorithm is based on clusterings and orderings and can reorder very large graphs quite fast. The experimental evaluation of this approach shows that combining the ordering produced by LLP with the WebGraph framework outperforms all currently known techniques, both for web graphs and for social networks. Some methods that claim to yield lower bits/edge ratios [45, 42, 98] do not address the issue of retrieving the edges fast. In [79] the authors introduce SlashBurn, an ordering method that offers the best bits per egde ratio according to the information theoretic lower bound, among other competing methods.

Buehrer and Chellapilla [42] exploit complete bipartite subgraphs (bicliques) on web graphs, i.e., groups of pages that share the same outlinks, and replace them with virtual nodes. However, the compression they achieve is not better than the compression of [29] while they also fail to be competitive speedwise, since they fall into the class of compression schemes rather than compressed data structures [30]. In computing the compressed size they do not take into account the offset, which, however, increases significantly with the increase of the compression ratio.

Clustering according to some meaningful measure naturally brings together nodes that are connected with the locality of reference or copy property. This is particularly useful in social networks where there is no apparent numbering of the nodes that brings them close to each other. In [98] the authors decompose the graph into small dense subgraphs, which can be represented more efficiently in terms of space. Their comparison with [29] is based on the naive approach of maintaining both the original

graph and its transposed version, whereas a more sophisticated approach, indicated in [30], outperforms them. In [70] the authors generalize on [42, 98], where the authors find bicliques and cliques respectively, and adapt clustering algorithms to find broader constructions that lie in between. They show that these more general dense subgraphs appear sufficiently more often than cliques and bicliques, thus designing a more general compact representation for them pays off.

Apostolico and Drovandi [13] visit the graph in a breadth-first fashion while compressing, and exploit *locality* and *similarity* by referencing the previous successor of the same node, or the successor of the previous list that is in the same ordinal position. The blocks of identical successors are recorded only once. However, their method is outperfomed by [30], and by a big margin as far as social network graphs are concerned.

In [116] Safro and Temkin present a multiscale approach for the network minimum logarithmic arrangement problem, i.e., the problem of finding an intrinsic ordering that optimizes directly the sum of the logarithms of the gaps (numerical difference between two successive neighbours). The resulting ordering may be used for graph compression if combined with a compression scheme like the WebGraph framework [29]. According to [30], some preliminary tests show that these orderings are promising especially on social networks; however, the implementation does not scale well for datasets with more than a few millions of nodes and so it is impractical for compressing large-scale graphs.

Our approach benefits from the reordering that results after applying *Layered Label Propagation* [30] on the input graph; we could also take advantage of other orderings such as the ones examined in [31, 79].

Figure 5.1 illustrates the aforementioned areas and the relations among them.

5.1.3 Organization

In this chapter we present the overview of our approach, along with some theoretical analysis and experimental evaluation on real datasets. In Section 5.2 we show and exploit the effect of locality in the compression of web and social network graphs. In particular, in 5.2.1 we identify the dense part of the graph based on the locality property, in 5.2.2 we present our compression algorithm and discuss its complexity and in 5.2.3 we evaluate our approach experimentally. In Section 5.3 we also perform data compression on part of the graph's elements, further improving the compression rate of our method. The data compression is described in 5.3.1 along with some theoretical analysis, and the new compression algorithm is presented in 5.3.2 and evaluated experimentally in 5.3.3.

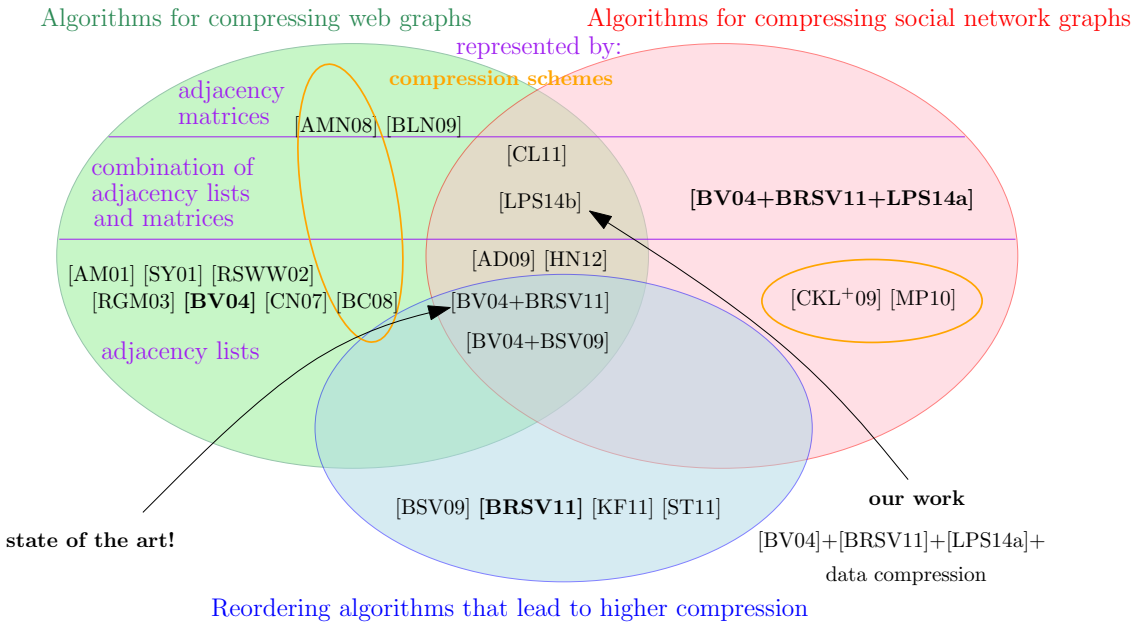
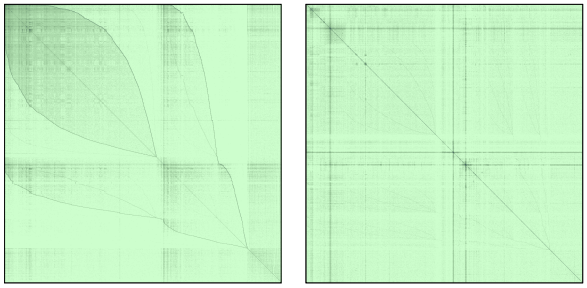


Figure 5.1: Related work



(a) (b)

Figure 5.2: `youtube-2007` before and after LLP.

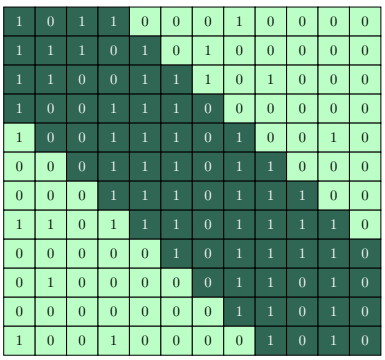


Figure 5.3: An adjacency matrix.

5.2 The effect of locality in compressing web and social network graphs

5.2.1 Identifying the dense part of the graph

Most compact graph representations are based either on the adjacency matrix representation [38] or on the adjacency lists representation [29] of the graph. For a given graph $G = (V, E)$ the adjacency matrix representation is preferred when G is dense, i.e., when $|E| = \Theta(|V|^2)$, while adjacency lists are preferred when G is sparse, i.e., when $|E| = O(|V|)$.

We combined the two kinds of representations after observing that social network

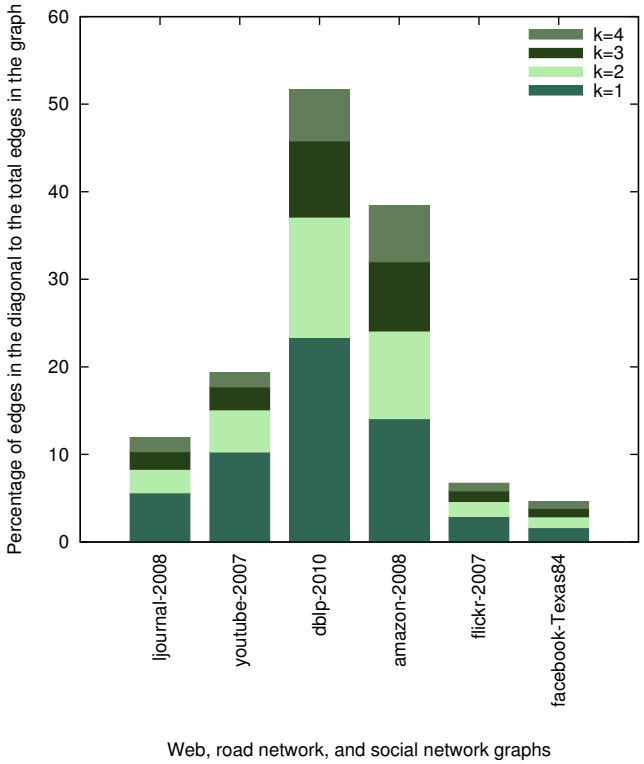


Figure 5.4: Percentage of edges contained in the diagonal stripe of various social network graphs for various stripe widths.

graphs, although rather sparse in general, have a dense part around the main diagonal of the graph's adjacency matrix after the LLP algorithm [30] has been applied on them. This tendency is shown in Figure 5.2, where the adjacency matrix of a graph from the `youtube` social network is illustrated before (5.2a) and after (5.2b) the reordering of its nodes.

More formally, we call this dense area the *diagonal stripe*, and define it as follows: let $k \in \mathbb{Z}_+$, an edge (i, j) is in the k-diagonal stripe, iff $i - k \leq j \leq i + k$. The 3-diagonal stripe of an example adjacency matrix is illustrated in Figure 5.3.

In the graphs we examined experimentally, a large number of edges tends to be in the diagonal stripe, meeting our expectations regarding the locality property. Figure 5.7 illustrates this trend for $k \in \{1, 2, 3, 4\}$ for the graphs of our dataset, described in detail in Section 5.2.3.1.

5.2.2 A hybrid method for graph compression

Having identified an opportunity to compress large parts of social network graphs effectively, we propose a hybrid method, which uses a bit vector to represent the diagonal stripe and resorts to the method in [29] to address the issue of compressing the re-

maining edges. For the rest of this chapter we will refer to our method as $\mathsf{BV}_{\mathcal{D}}$ and to the method in [29] as BV.

Every possible pair of nodes (a, b) lying in the diagonal stripe is mapped through a simple function to the bit vector. Thus, the existence of an edge there can be verified in constant time. A big percentage of these pairs represent edges absent from the graph. However, including those pairs in our representation allows us to be aware of the position of every pair and not resort to using an index as in [38], which would not only introduce a similar space overhead, but would dramatically increase the retrieval time as well.

By using BV to compress the rest, sparse part, of the graph, we manage to provide a full graph compression framework and perform comparisons over the whole graph, not only the diagonal stripe. The computational complexity of this approach is approximately equal to the complexity of BV alone, as mapping the diagonal stripe to a bit vector is linear in the number of diagonal edges. Furthermore, this mapping can only decrease the query time on the compressed graph's elements, when compared with the query time of BV alone.

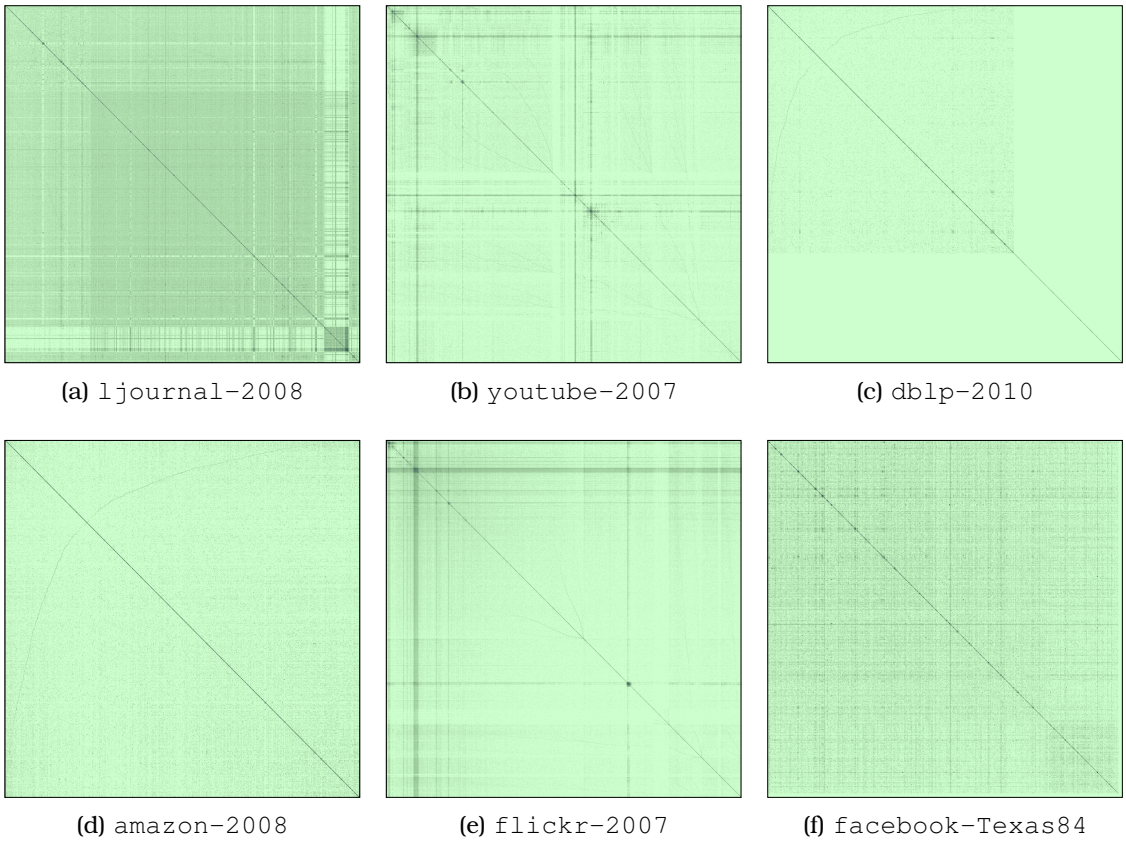
5.2.3 Experimental evaluation

5.2.3.1 Dataset

In order to test our approach we used a dataset of six social network graphs. Figure 5.5 provides an illustration of their adjacency matrices, where one can clearly see how the diagonal stripe stands out in almost all of the graphs. The origin and characteristics of our graphs are summarized in the following list:

- `ljournal-2008`: *LiveJournal* is a virtual community social website that started in 1999. It comprises $5,363,260$ nodes and $79,023,142$ edges.[1]
- `youtube-2007`: *Youtube* is a video-sharing website that includes a social network. It comprises $1,138,499$ nodes and $5,980,886$ edges.[3]
- `dblp-2010`: *DBLP* is a bibliography service. Each vertex represents an author, and an edge links two vertices if the corresponding authors have collaborated. It comprises $326,186$ nodes and $1,615,400$ edges.[2]
- `amazon-2008`: *Amazon* is a symmetric graph describing similarity among books as reported by the Amazon store, comprising $735,323$ nodes and $5,158,388$ edges.[6]
- `flickr-2007`: *Flickr* is a photo-sharing website based on a social network. It comprises $1,715,255$ nodes and $31,110,082$ edges.[3]

[1] Collected in [45], retrieved by LAW: `http://law.di.unimi.it/`
[2] Collected by LAW: `http://law.di.unimi.it/`
[3] Part of the IMC 2007 datasets with LLP [30] applied on it (`http://socialnetworks.mpi-sws.`



(a) ljournal-2008 (b) youtube-2007 (c) dblp-2010

(d) amazon-2008 (e) flickr-2007 (f) facebook-Texas84

Figure 5.5: Visualizations of the adjacency matrices of some social network graphs.

- facebook-Texas84: *Facebook* is the most successful online social networking service. Its Texas84 subnetwork comprises $36,371$ nodes and $3,181,310$ edges.[4]

The aforementioned graphs vary in size and cover a wide range of social networking services. Thus, they form a thorough evaluation environment for our proposed method.

5.2.3.2 Compression ratio comparison

Table 5.3 shows the number of nodes and edges in each graph, the percentage of edges in the diagonal stripe, the compression ratio achieved by the BV technique [29], and the one achieved by our proposed method ($\mathsf{BV}_{\mathcal{D}}$) for a given k.

As expected, the largest improvement (10%) was achieved for dblp-2010, which has the densest diagonal among all graphs in our dataset. Notable improvements were also observed for graphs youtube-2007 (3%) and amazon-2008 (2%). Surprisingly, for the other three graphs, viz., ljournal-2008, flickr-2007 and facebook-Texas84, $\mathsf{BV}_{\mathcal{D}}$ also managed to surpass the performance of BV, even though the percentage of

org/data-imc2007.html).

[4]The largest of the Facebook100 graphs containing friendships from 100 US universities in 2005 (https://archive.org/details/oxford-2005-facebook-matrix).

Table 5.1: Comparison with BV method.

graph	# nodes	# edges	% of edges in diagonal	k	compression ratio (bits/edge)	
					BV	$BV_{\mathcal{D}}$
ljournal-2008	5,363,260	79,023,142	5.62%	1	11.84	11.80
youtube-2007	1,138,499	5,980,886	15.10%	2	14.18	13.79
dblp-2010	326,186	1,615,400	37.12%	2	8.63	7.76
amazon-2008	735,323	5,158,388	43.56%	5	10.77	10.56
flickr-2007	1,715,255	31,110,082	4.66%	2	9.81	9.76
facebook-Texas84	36,371	3,181,310	3.84%	3	8.82	8.80

edges in their diagonal stripes is relatively small.

By outperforming BV for all the graphs in our dataset, we proved that the effect of our observations, even when utilized with a simple approach such as that of $BV_{\mathcal{D}}$, is very powerful on social network graphs.

5.2.3.3 The effect of parameter k

Achieving a good compression ratio with $BV_{\mathcal{D}}$ depends heavily on choosing an appropriate width for the diagonal stripe of the given graph, defined by k. The optimal values of k for the graphs of our dataset are illustrated in Table 5.3.

As k increases, more and more edges are included in the diagonal stripe, which, however, becomes progressively sparser. We have found that a good selection of value for this parameter ranges between 1 and 5. The most appropriate value can only be known a posteriori, as it depends on the exact structure of the graph and does not only determine the bits per edge ratio of the diagonal part, but also the compression ratio of the subgraph compressed with BV. However, our results indicate that improvement over BV occurs for most of the values within this range; e.g., for graphs `dblp-2010` and `amazon-2008`, better results were achieved for $k \in [1, 7]$ and $k \in [1, 8]$ respectively.

5.3 Pushing the envelope in graph compression

5.3.1 Overview of our approach

This section presents our approach for compressing directed graphs. Our approach builds on and improves the Boldi and Vigna compression method of the *WebGraph framework* [29]. For the sake of simplicity, we will refer to the Boldi and Vigna method as BV throughout the section. We first review the BV techniques (Section 5.3.1.1), then we isolate a dense subgraph of the input graph (Section 5.3.1.2), in particular a

stripe around its main diagonal, and provide an explanation of how we can compress it separately along with a theoretical analysis of this approach (Section 5.3.1.3).

5.3.1.1 The Boldi et al. techniques

In [29], Boldi and Vigna propose a number of techniques that exploit *locality* and *similarity*, two properties that are known to appear in the links of a web graph [112].

The adjacency lists of a graph are pictured firstly using a modified gap representation, that utilizes the *locality* property, and then as bit vectors, named *copy lists*, that take advantage of the fact that the adjacency lists share large subsequences of edges (*similarity* property). *Copy lists* are further compressed with a variation of run-length encoding, since they tend to contain runs of 0s and 1s.

The number of previous adjacency lists that are examined in order to discover possible reference lists is called *window*, and its size poses a tradeoff between compression ratio and compression/decompression time. The maximum reference count is a second parameter used by this scheme, that imposes a limit on the lengths of reference chains.

The remaining extra nodes exhibit consecutivity as well. Hence, integer intervals are used for their compression, but only for the subsequences that correspond to intervals whose length is not below a certain threshold (L_{min} in [29]). The list of the *residuals* (remaining integers) is compressed in a differential manner.

In [30], Boldi et al. apply a reordering algorithm that brings the graph to a state where the aforementioned properties can be further exploited.

5.3.1.2 Exploiting the dense part of the graph

We improve the state-of-the-art algorithm BV for the compression of web graphs [29], by proposing to store separately the denser part of the graph, i.e., the part of the graph corresponding to the edges that are close to the main diagonal of the graph's adjacency matrix.

Graphs created by human activity usually possess the locality of reference property and the copy property, which are surfaced after applying LLP [30] on them, or any other clustering technique, e.g., [31, 79], that permutes the graph in a similar fashion. We exploit these properties to improve the compressed data structure. Due to the above properties, an edge is with high probability *close* to the main diagonal of the adjacency matrix representing the graph. Hence, given that these graphs are generally rather sparse due to the power law distributed nodes' degrees (indegrees and outdegrees) [58], the graph corresponding to the main diagonal and the area around it is denser than the rest of the graph. We call this area the *diagonal stripe*, and formally define it as follows:

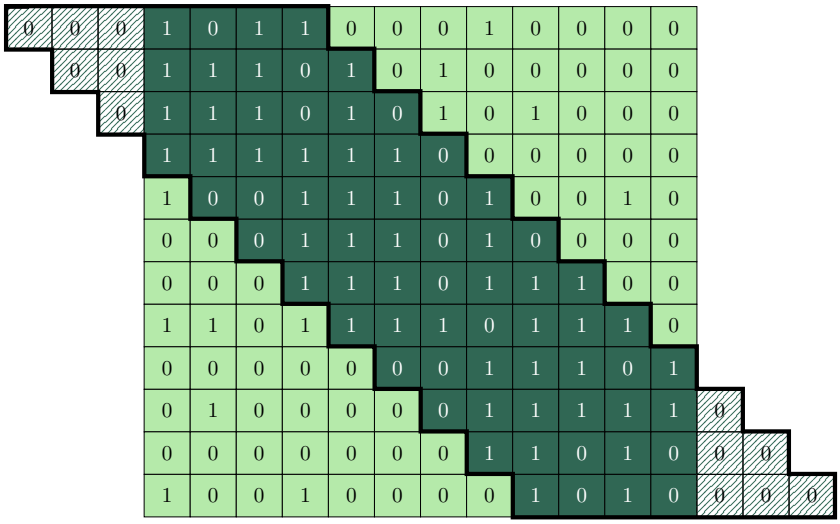


Figure 5.6: Example of an adjacency matrix.

Definition 4. For a graph $G = (V, E)$ and $k \in \mathbb{Z}_+$, the k-diagonal stripe of G comprises the following set of entries: $\{(i, j) \mid i - k \leq j \leq i + k \text{ and } i, j \in \{0, \ldots, |V|\}\}$.

To illustrate, in Figure 5.6 we present within a bold line the 3-diagonal stripe of an example adjacency matrix.

In the graphs we examined experimentally, large number of edges tend to be in the diagonal stripe, meeting our expectations regarding the locality of reference property. This trend for $k \in \{1, 2, 3, 4\}$ for the graphs of our dataset, described in detail in Section 5.3.3.1, is illustrated in Figure 5.7.

As k increases, the bits/edge ratio required to store the stripe increases as well. The density of edges in the diagonal stripe decreases as we are moving farther away from the diagonal, where edges are met less frequently. However, even a stripe of smaller density is sometimes useful, as it may lead to higher compression of the whole graph.

The computation of the bits/edge needed to represent the diagonal stripe is straightforward if we know the percentage of edges in it: Consider $k \in \mathbb{Z}_+$ and a graph $G = (V, E)$ with a percentage p of E belonging in the k-diagonal stripe. These edges are represented with $\frac{(2k+1)|V|}{p|E|}$ bits/edge. For example, graph `roadNet-PA` of our dataset, described in Section 5.3.3.1, has 68.41% of its edges in the 7-diagonal. Therefore these edges are represented with $\frac{15|V|}{0.6841|E|} = 7.76$ bits/edge.

Isolating the diagonal stripe in a way similar to [94] and compressing the rest of the graph as explained in [29], achieves better compression than using the method presented in [29] alone, as we later demonstrate experimentally (Section 5.3.3). Table 5.2 presents the compression ratio achieved by BV for the remaining graph for various diagonal stripe widths, which is essentially the lower bound of our overall compression.

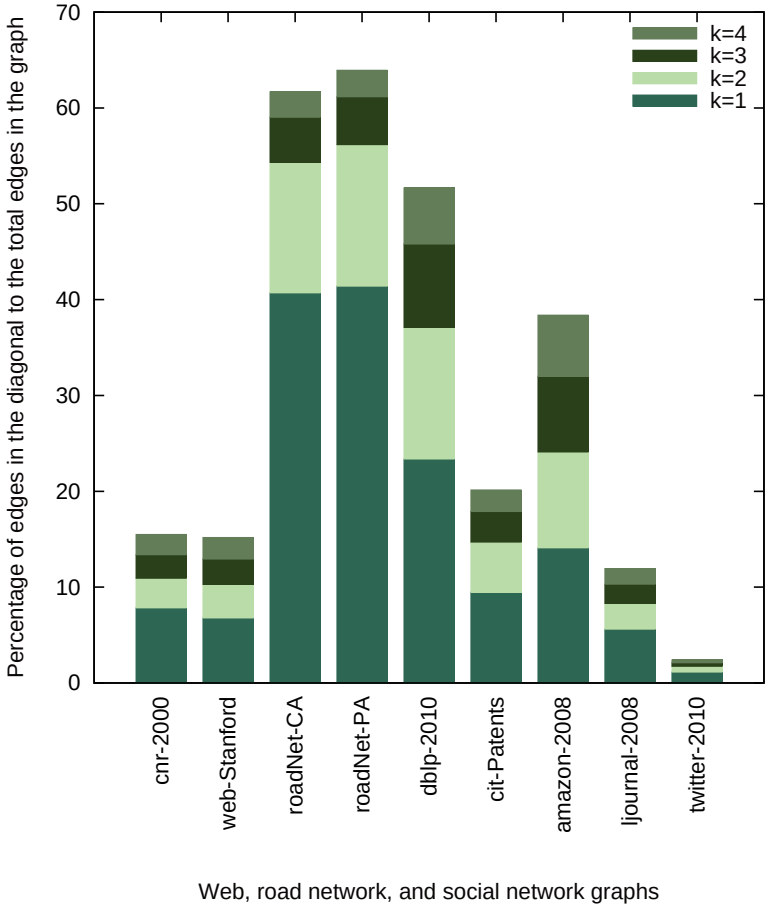


Figure 5.7: Percentage of edges contained in the diagonal area of web, road network and social network graphs.

5.3.1.3 Compressing the diagonal stripe

In this section we describe the motivation and sketch the techniques behind isolating a dense subgraph of the input graph, in particular a stripe around its main diagonal, and compressing it separately. We also present some theoretical analysis for our proposed approach.

Adjacency matrix format In order to exploit the high concentration of edges in the diagonal stripe, and, thus, take advantage of the locality of reference property, we store it separately from the rest of the graph in the format of an adjacency matrix. We opted for the adjacency matrix representation of the diagonal as the high concentration of edges in the diagonal forms a dense graph. Formally, a graph $G = (V, E)$ is dense if $|E| = \Theta(|V|^2)$. For example, as shown in Figure 5.8, an edge is more likely to exist in the part of the adjacency matrix corresponding to the diagonal stripe, or even in the part around the stripe, than far away from it. This is exactly the case in the graphs

Table 5.2: Lower bound of our compression: The bits/edge required by BV for the graph apart from the diagonal stripe.

graph	BV	***lower bound of our compression***			
		k = 1	k = 2	k = 3	k = 4
cnr-2000	3.71	3.37	3.28	3.22	3.16
web-Stanford	4.06	3.76	3.63	3.56	3.48
roadNet-CA	13.30	10.39	9.31	8.89	8.58
roadNet-PA	12.86	10	8.85	8.41	8.09
dblp2010	8.63	7.47	6.75	6.37	6.02
cit-Patents	14.72	14.21	13.88	13.67	13.51
amazon-2008	10.77	10.32	9.93	9.62	9.29
ljournal-2008	11.84	11.60	11.47	11.39	11.31
twitter-2010	14.52	14.41	14.35	14.32	14.29

that we are dealing with here.

Data compression Using data compression techniques that exploit the redundancy of the diagonal stripe, represented by an adjacency matrix as described above, allows us to reduce the size of the stripe significantly. Shannon's source coding theorem states that it is impossible to compress with an average number of bits per symbol less than the entropy of the source. We present a proposition that imposes an upper bound to that limit, and provides us with an estimation of the space requirements of our method for the dense part of the graph. Comparing this estimation for various widths of the diagonal stripe of a graph, to the compression ratio of the state-of-the-art method, allows us to assess the overall room for improvement and the optimal width of the stripe. However, the estimation on the latter is far from accurate due to the delicate balance between easing the task of compressing the rest of the graph by including as many edges as possible in the diagonal stripe and minimizing its ratio.

Proposition 6. *Consider $k \in \mathbb{Z}_+$ and a graph $G = (V, E)$ with a percentage p of its edges belonging in the k-diagonal stripe. The minimum expected compression ratio of the diagonal stripe is upper bounded by $\frac{\log \binom{(2k+1)|V|}{p|E|}}{p|E|}$ bits/edge.*

Proof. The diagonal stripe consists of $(2k+1)|V|$ bits and exactly $p|E|$ of them represent edges. We model the stripe as a random variable $X \in \{0,1\}^{(2k+1)|V|}$.

Shannon's source coding theorem states that the minimal possible expected length of codewords, which in our case is the best attainable compressed size of the diagonal stripe, is no less than the entropy of the input word (diagonal stripe) [118].

The entropy of X is $H(X) = -\sum_{i=1}^{n} p_i \log p_i$, where n is the number of all possible diagonal stripes and p_i is the probability of stripe i. As $n = \binom{(2k+1)|V|}{p|E|}$, and assuming

that all possible stripes are equally likely, the maximum entropy becomes

$$H(X) = \log \binom{(2k+1)|V|}{p|E|}.$$

According to Shannon, the minimal expected wordlength S is $\mathbf{E}[S] = H(X)$. Thus, since we have $p|E|$ edges, the minimum expected compression ratio is

$$\frac{H(X)}{p|E|} = \frac{\log \binom{(2k+1)|V|}{p|E|}}{p|E|}.$$

□

For example, for the graph `roadNet-PA` the upper bound of the minimum expected compression ratio of the 7-diagonal stripe is $\frac{\log \binom{15|V|}{0.6841|E|}}{0.6841|E|} = 2.98$ bits/edge.

Minimizing the compression ratio with techniques such as Huffman or Arithmetic coding [123] may have a negative impact on the time needed to access the elements of the graph, as we will then need to decompress large parts, or even the whole diagonal stripe, in order to answer simple queries. We wish to retain the ability to access the elements of the stripe in constant time after compressing them. Thus, we encode them using a form of lossy, but fixed-length encoding to preserve the direct access of the edges.

5.3.2 Compressing the graph

This section presents BV+, an algorithm for compressing directed graphs that is the product of our line of thinking in Section 5.3.1. BV+ is outlined in Algorithm 1. BV+ receives as input a directed graph $G = (V, E)$, and parameters k and b, and gives as output a compressed representation of G.

As a first step, the algorithm constructs the k-diagonal stripe of graph G and the set of all edges that do not belong in the k-diagonal stripe (lines 2-8). The k-diagonal stripe can be considered as an array of bit arrays where each bit array consists of exactly $2k + 1$ elements and corresponds to a row of the diagonal stripe as illustrated in Figure 5.8. The value of an element equal to 1 signifies the presence of an edge. Likewise, the value of an element equal to 0 signifies the absence of an edge. The first and last rows in the k-diagonal stripe are complemented with 0s to fix the number of $2k + 1$ elements for each row.

Algorithm 1: BV+(G, k, b)

input : A directed graph $G = (V, E)$, and parameters k and b.
output: A compressed representation of G.

1 **begin**
2 setNonD ← set();
3 k-diagonalStripe ← array(array($\underbrace{[000\ldots0]}_{\text{2k+1 bits}}$) × $|V|$);
4 **foreach** $(u, v) \in E$ **do**
5 **if** $u - k \leq v \leq u + k$ **then**
6 k-diagonalStripe$[u][v] \leftarrow 1$;
7 **else**
8 setNonD ← setNonD ∪ (u, v);
9 seqDict ← dict();
10 **foreach** *seq* ∈ *k-diagonalStripe* **do**
11 **if** *seq* ∉ *seqDict* **then**
12 seqDict[seq] ← 1;
13 **else**
14 seqDict[seq]++;
15 **foreach** *(key, value)* ∈ *seqDict* **do**
16 seqDict[key] ← value × # of 1s ∈ key;
17 seqDict ← sort seqDict by value (desc. order);
18 seqSet ← {first $2^b - 1$ sequences (keys) of seqDict};
19 **foreach** *seq* ∈ *k-diagonalStripe* **do**
20 **if** *seq* ∈ *seqSet* **then**
21 *use b bits to compress seq*;
22 **else**
23 bestSeq ← bestSubset(seqSet, seq, k);
24 *use b bits to compress bestSeq*;
25 setNonD ← setNonD ∪ {edges of seq that were left out of bestSeq};
26 *compress setNonD using* BV;

Diagonal stripe compression As already mentioned, the k-diagonal stripe consists of $|V|$ rows where each row comprises $2k + 1$ elements. We create a dictionary to hold information about our rows (line 9). Every row, that is, every sequence of $(0, 1)$-elements, yields an integer value. We have empirically observed that the frequencies of these integer values tend to follow a power law distribution, so we decided to pick the values that contain the higher volume of information.

We iterate over the set of rows, and store each distinct $(2k + 1)$-bit array along with its frequency in the set of rows in the k-diagonal stripe in our dictionary (lines 10-14). We do not rest on using the $(2k+1)$-bit arrays met most frequently among the rows, but we also take into account the number of edges they represent. More explicitly, let us suppose we had used $k = 3$ and we had observed the sequences 0001000 and 0010110 occurring 500 and 300 times respectively. While the first sequence is more frequent, it fails to represent more than one edge. Hence, the second sequence is in fact preferable.

Function: bestSubset(seqSet, seq, k)

input : A set of candidate sequences, seqSet, parameter k, and a given sequence seq.
output: Best available sequence bestSeq.

```
1  begin
2      bestSeq ← array(underbrace([000...0], 2k+1 bits));
3      max ← 0;
4      foreach candidateSeq ∈ seqSet do
5          counter ← 0;
6          flag ← False;
7          foreach i ← 1 to 2k + 1 do
8              if candidateSeq[i] = 0 and seq[i] = 1 then
9                  flag ← True;
10             if candidateSeq[i] = seq[i] = 1 then
11                 counter++;
12         if counter ≥ max and !(flag) then
13             bestSeq ← candidateSeq;
14             max ← counter;
15     return bestSeq;
```

The multiplication of the frequencies with the number of bits that are set guarantees that the chosen representations will not only occur often in the particular diagonal, but will represent a significant amount of edges as well, thus minimizing the overall bits per edge ratio (lines 15-16).

Then, we choose an integer number b of bits to use for their representation and represent only the $2^b - 1$ most appropriate of these sequences, each one using a binary number of b bits. This is done by sorting these sequences by the product of their frequency times the number of edges each one contains (line 17), i.e., the number of 1s in their binary representation, and then picking the first $2^b - 1$ of them (line 18). In this way, we make sure that we will pick not just the most frequent values in the diagonal, but also the most important ones, because picking representations that hold a limited number of edges would lead to a waste of bits. The role of b is to determine the channel capacity, and with it, the loss rate of our scheme. The optimal value for b is highly dependent on the distribution that the frequencies of the values follow. We keep one b-bit binary number to denote the absence of edges in a specific row of the diagonal stripe.

As a final step, we iterate again over the set of rows of the k-diagonal stripe, and for each row of the diagonal stripe we use its compressed representation if the corresponding sequence exists among the ones picked in the aforementioned step (lines 20-21), or the compressed representation of the *best available* sequence and add the missing edges to the set of edges that do not belong in the k-diagonal stripe (lines 22-25). In the

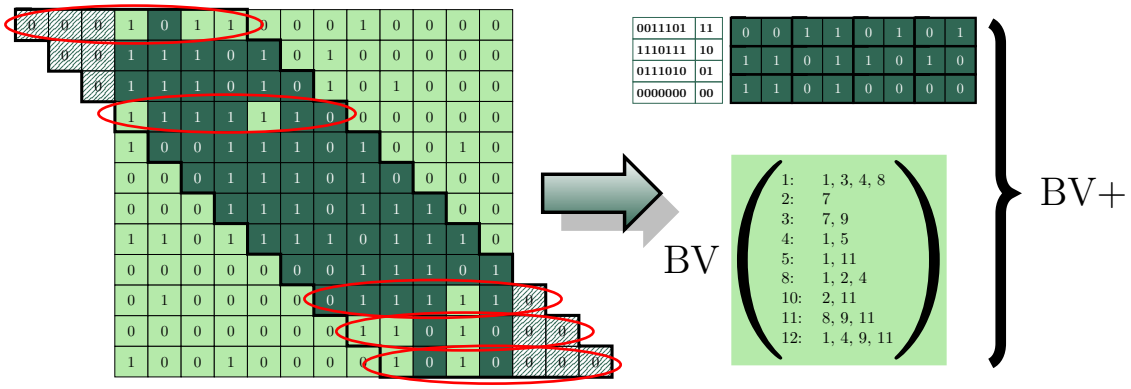


Figure 5.8: Compressing the graph with BV+.

latter case, by *best available*, we denote the sequence that has the most $1s$ in the same position with the sequence in question from the ones picked, and does not have a single 1 where this sequence has a 0. The best available sequence is provided by function bestSubset, which receives as input a set of candidate sequences, parameter k, and a given sequence. Function bestSubset first initializes a best sequence candidate, represented by a bit array, with $2k+1$ number of 0s (line 2), and it also initializes a counter that holds the number of 1s that two sequences have in common (line 3). The best available sequence, i.e., the one whose set of positions of 1s in it is the best (maximal) subset of the sequence in question, is then calculated by iterating the set of candidate sequences and performing some checks (lines 4-14), and, finally, returned (line 15). Note that even if no best available sequence is found among the set of candidate sequences, the initialized best available sequence candidate (line 2) will be returned. In the aforementioned example, suppose that the second sequence, viz., 0010110, was indeed elected among the 2^b-1 ones, while another sequence 0011110 was not. We utilize their similarity by using the representation of the former one for the latter one as well, and manage to capture 3 of its 4 edges without extra cost. In the upper right part of Figure 5.8 we can see the mapping of the selected values to their b-digit representation and the compressed diagonal that results after applying the actions described above.

The edges that are excluded during this step are added to the ones existing outside of the diagonal stripe. These edges will be then compressed using BV, thus, our overall method is lossless. In Figure 5.8 the edges of the diagonal that are compressed as described above are contained in dark grey cells, while those that are compressed using BV are in light grey cells. When every row has been substituted with a compressed representation, the compressed diagonal is ready.

As a result of using the above procedure, for the graph `roadNet-PA` the compression ratio of the 7-diagonal stripe with b set to 2 is 1.95 bits/edge. However, even

a compression ratio greater than the upper bound of the minimum expected (2.98 bits/edge as obtained in the example in Section 5.3.1.3 for `roadNet-PA`), and in any case smaller than that of the uncompressed graph, would be acceptable as we do not need to decompress the whole stripe to access the desired edges, in contrast to more compact entropy encoding approaches, such as Huffman or Arithmetic coding [123].

Non Diagonal part compression The final step for algorithm BV+ is to compress the remaining edges, i.e., the edges that initially belonged outside of the k-diagonal stripe (line 8), together with the edges that were left out during the compression of the diagonal stripe (line 25), with the BV method (line 26). As shown in Figure 2, the output of BV+ is the lossy compressed representation of the diagonal stripe plus the output of BV for the remaining elements. Besides using exclusively the BV method, the straightforward nature of our approach and the structure of our algorithm makes it an attractive technique that any compression scheme can benefit from.

5.3.2.1 Size of the compressed graph

Let n be the number of nodes in the graph (i.e., $n = |V|$) and b be the parameter that ultimately defines the width of a compressed representation of the diagonal stripe, as described earlier. The size of the compressed graph is equal to $bn+S_{BV}$ bits, where S_{BV} is the size of the set of edges that are outside of the diagonal stripe plus the set of edges that were left out of the fixed length encoding of the diagonal (during the compression of the diagonal), compressed using the BV algorithm.

5.3.2.2 Time Complexity

Here, we discuss the time complexity of BV+ and compare it to the complexity of BV. The diagonal stripe is compressed in linearithmic time in the worst case ($O(n \log n)$) and the remaining edges in time less than with BV, as they form a graph that is smaller than the initial one.

- The time complexity of verifying the existence of a specific edge is $O(1)$ if the edge belongs in the *compressed* diagonal stripe, and less than that of BV otherwise[5].
- The time complexity of retrieving all neighbours of some node is $O(b)$ for the neighbours that belong in the *compressed* diagonal stripe, and less than that of BV for the rest of the neighbours[5].

The efficiency of our approach benefits from the existence of multi-core processors, since in any multi-core (e.g., 2-core) machine the aforementioned queries in and outside

[5]Since edges have to be retrieved from a compressed by BV graph, which is initially smaller than the input graph.

Table 5.3: Comparison with BV method.

graph	*# nodes*	*# edges*	*# edges in compressed diagonal*	*compression ratio (bits/edge)*		BV+ *parameters*	
				BV	BV+	*k*	*b*
cnr-2000	325,557	3,216,152	194,639	3.71	3.62	17	2
web-Stanford	281,903	3,985,272	270,220	4.06	3.90	1	2
roadNet-CA	1,965,206	5,533,214	3,569,145	13.30	10.58	7	6
roadNet-PA	1,088,092	3,083,796	2,062,741	12.86	10.07	7	6
dblp-2010	326,186	1,615,400	928,702	8.63	7.2	24	7
cit-Patents	3,774,767	33,037,894	6,303,138	14.72	14.25	9	6
amazon-2008	735,323	5,158,388	3,057,268	10.77	10.07	23	15
ljournal-2008	5,363,260	79,023,142	6,045,619	11.84	11.78	2	4
twitter-2010	41,652,230	1,468,365,182	37,906,525	14.52	14.42	17	6

the diagonal stripe take place in parallel, thus, the makespan is the longer among the two tasks.

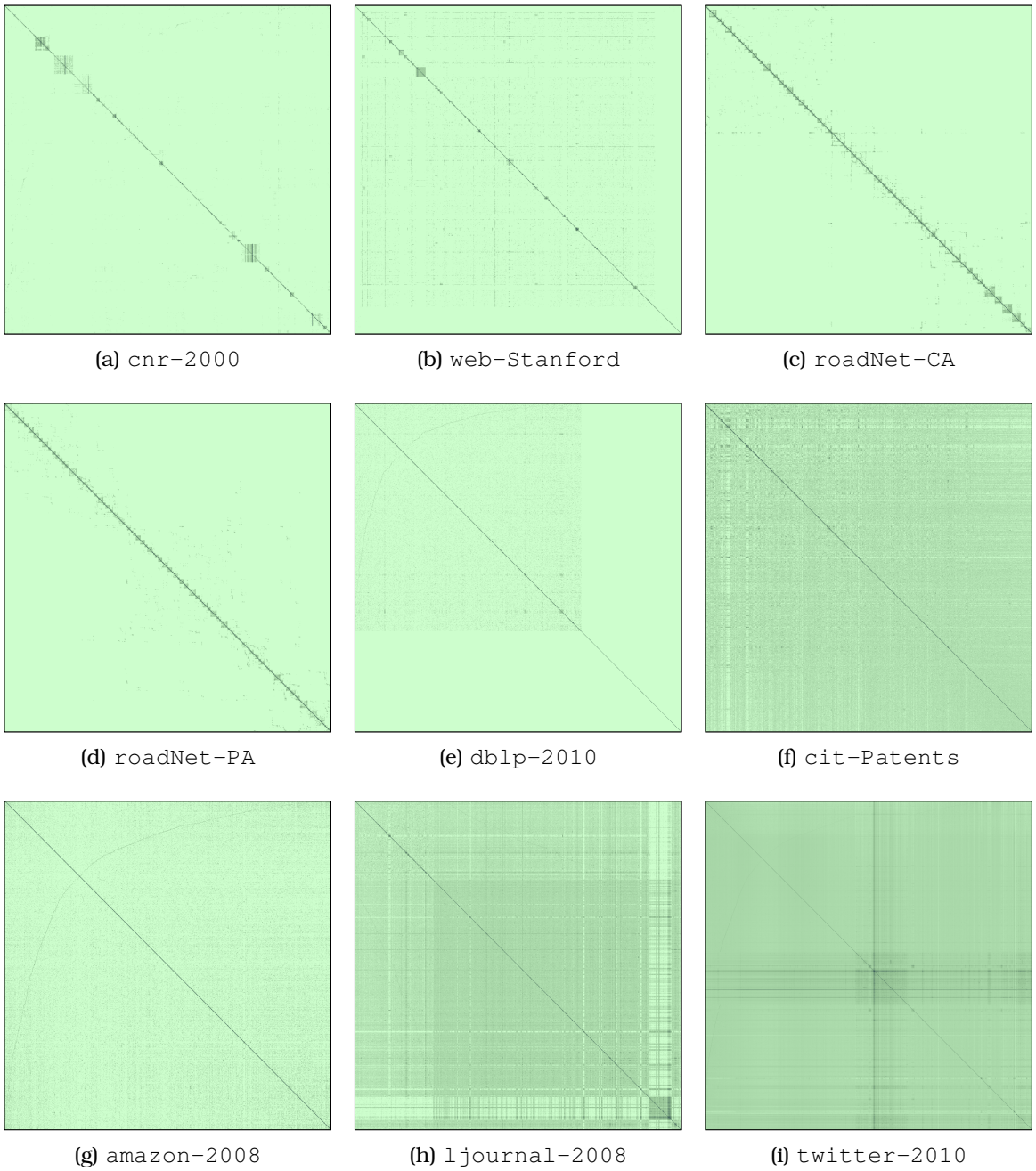
We infer that BV+ outperforms BV in terms of time needed for searching/retrieving edges in a graph compressed using any one of them, and we also show this experimentally later in Section 5.3.3.

The high compression of BV has a negative impact on the time needed to access some of the graph's elements: the retrieval of the incoming edges of a specific node becomes involved [29]. BV+ induces an improvement in this aspect, as part of the graph's edges, i.e., the part that belongs in the diagonal stripe, is accessed in constant time.

5.3.3 Experimental evaluation

We implemented and tested our approach on a wide variety of large scale graphs. We list below the technical specifications of the machine used for implementing and testing our algorithm. In Section 5.3.3.1 we describe the dataset used for our experiments. We present the results of our experiments, i.e., compression ratios and times, in Sections 5.3.3.2 and 5.3.3.3 respectively, and discuss the role of the algorithm's input parameters in Section 5.3.3.4.

We implemented and ran algorithm BV+ using OpenJDK 7 build 25, which implements Java SE 7; our code is available upon request. The experiments were carried out on a computer with an Intel®Core™ 2 Duo CPU E8400 with a CPU frequency of 3.00GHz and a 6MB L2 cache, a total of 8GB DDR2 800MHz RAM, a SATA3 Intel SSD hard disc of 80GB, and the Linux Mint 13 (Maya) x86 64 OS. Only one of the CPU cores was used for the experiments.



(a) cnr-2000 (b) web-Stanford (c) roadNet-CA

(d) roadNet-PA (e) dblp-2010 (f) cit-Patents

(g) amazon-2008 (h) ljournal-2008 (i) twitter-2010

Figure 5.9: Heat maps of the adjacency matrices of web (a, b), road network (c, d), citation (e, f), and social network (g, h, i) graphs.

5.3.3.1 Dataset

The dataset that we used to apply and test our compression technique, comprises nine well-studied [29, 30, 93, 45, 98] web, road network, citation, and social network graphs. Figure 5.9 provides an illustration of their adjacency matrices, where one can clearly see how the diagonal stands out for all the graphs. The origin and characteristics of our graphs are summarized in the following list:

- `cnr-2000`: a web graph (directed) from a crawl of the Italian CNR domain. It comprises $325,557$ nodes and $3,216,152$ edges.[6]
- `web-Stanford`: a web graph from Stanford University, collected in 2002. It comprises $281,903$ nodes and $3,985,272$ edges.[7]
- `roadNet-CA`: the road network of California (undirected). It comprises $1,965,206$ nodes and $5,533,214$ edges.[7]
- `roadNet-PA`: the road network of Pennsylvania (undirected). It comprises $1,088,092$ nodes and $3,083,796$ edges.[7]
- `dblp-2010`: an undirected scientific collaboration network graph from the DBLP bibliography service. Each vertex represents an author and an edge links two vertices if they have worked together. It comprises $326,186$ nodes and $1,615,400$ edges.[6]
- `cit-Patents`: a citation graph which includes all citations made by U.S. patents granted between 1975 and 1999. It comprises $3,774,767$ nodes and $33,037,894$ edges.[7]
- `amazon-2008`: a symmetric graph describing similarity among books as reported by the Amazon store. It comprises $735,323$ nodes and $5,158,388$ edges.[6]
- `ljournal-2008`: LiveJournal [8] is a virtual community social site started in 1999; in this social network friendship is non-symmetric so the graph is directed. It comprises $5,363,260$ nodes and $79,023,142$ edges.[9]
- `twitter-2010`: Twitter is a social networking and microblogging service; To the best of our knowledge, this is the largest available social network graph. It comprises $41,652,230$ nodes and $1,468,365,182$ edges.[10]

The aforementioned graphs vary in category, size, and type, and are therefore very good candidates for examining the effectiveness of our proposed method.

[6] Collected by LAW: `http://law.di.unimi.it/`

[7] Collected by SNAP: `http://snap.stanford.edu/snap/`

[8] `http://www.livejournal.com/`

[9] Collected in [45], retrieved from LAW: `http://law.di.unimi.it/`

[10] Collected in [92], retrieved from LAW: `http://law.di.unimi.it/`

[11] In [98] the algorithm provides both predecessors and successors but a simple strategy proposed in [30] indicates that less than double bits per link are needed for a fair comparison.

Table 5.4: Comparison with other methods.

method	*graph*	*compression ratio (bits/edge)*	
		other	BV+
[13]	ljournal-2008	14.97	11.78
	amazon-2008	12.39	10.07
	dblp-2010	7.47	7.2
[98][11]	web-Stanford	9.88	3.90
	cit-Patents	25.69	14.25

Table 5.5: Access times (in ns) for a web, a road network, and a social network graph.

graph		cnr-2000	roadNet-CA	ljournal-2008
Edge Exists	**BV+ (D)**	645	600	663
	BV+ (Non-D)	2,286	832	4,373
	BV+ (total)	2,187	728	4,089
	BV	2,397	923	4,518
Successors	**BV+ (D)**	643	668	627
	BV+ (Non-D)	1,947	849	1,940
	BV+ (total)	1,868	768	1,840
	BV	2,159	891	2,009

5.3.3.2 Compression ratio comparison

We compared our approach with BV as well as with other graph compression methods. The results are outlined in Tables 5.3 and 5.4 respectively.

Table 5.3 shows the number of nodes and edges of each graph, the compression ratio achieved by the BV technique [29], and the one achieved by our proposed method (BV+) for all the graphs tested. The number of edges that was ultimately included in the compressed diagonal stripe and its parameters are also displayed.

For the two web graphs, we observed an improvement of 2.4% for `cnr-2000` and 3.9% for `web-Stanford`. Our method attained impressive results for the two road network graphs. An improvement of 20.5% and 21.7% was achieved with the use of BV+ for `roadNet-CA` and `roadNet-PA` respectively. Regarding the citation and social network graphs, our method had a large impact on `dblp-2010` and `amazon-2008`, achieving a 16.6% and a 6.5% compression ratio improvement over the compression ratio of BV respectively, and offered smaller, but significant nonetheless, compression ratio improvements for the other three graphs.

The higher levels of compression for certain graphs is due to the fact that a higher percentage of their edges exist in the diagonal stripe, as opposed to the rest of the graphs. However, even with the less intense clustering, our technique manages to reduce the compression ratio of BV significantly. As it can be seen in Figure 5.7, these

graphs are `roadNet-CA`, `roadNet-PA`, `dblp-2010`, and `amazon-2008`. The somewhat smaller impact of BV+ on the compression ratio of `web-Stanford`, `cnr-2000`, `cit-Patents`, `twitter-2010`, and even `ljournal-2008` is due to the fact that BV does not leave enough room for improvement, as Table 5.2 illustrates. That is, the remaining edges that are assigned to BV, occupy most of the final compressed file. The lower bound essentially signifies the compression that we would be able to achieve if we could represent the stripe using 0 bits/edge. For the latter graphs the lower bound does not deteriorate at a satisfying rate as k increases, i.e., it remains almost stable.

In case our method provided no improvement for a given graph, it could easily fall back to BV instead of BV+ by setting b to zero.

Table 5.4 shows the comparison of our approach to two other recent methods [13, 98]. The methods were examined in [30] and were shown to be inferior than BV, but we chose to include this comparison for reasons of completeness.

We also evaluated BV+ for graphs of our dataset after SlashBurn [79] had been applied on them. Even though the reordering of SlashBurn favoured BV+ over BV for the graphs tested, the compression ratio achieved using this representation was significantly larger, as exploiting SlashBurn does not seem to aid the compression techniques of BV.

5.3.3.3 Access time comparison

Table 5.5 presents the results obtained by the comparison of BV and BV+ as far as access times are concerned. We tested the responsiveness of the two methods when asking if a node is a successor of another one (*Edge Exists*), and when inquiring all the successors of a node (*Successors*).

For the former query, we searched for every edge present in the graph and calculated the mean average of those that were held in the compressed diagonal representation and of those that were compressed using the BV method. In the first case, a simple test –if the corresponding bit of the uncompressed diagonal representation is set– is enough. In the second case, we iterate over the successors of a node using a *LazyIntIterator*, until we find the edge or run out of them.

For the latter query, we asked for the successors of all nodes of each graph in Table 5.5 and calculated the mean average time of the responses. The built-in *successors()* method of class *BVGraph* was used for the part of the graph that was compressed with the BV method. For the rest of the edges, i.e., the edges belonging in the compressed diagonal stripe, a list was populated by adding the successors whose corresponding bits in the original diagonal stripe are set.

We took into account the percentage of edges existing in the diagonal of each graph

to the total edges of the graph (calculated using the data in Table 5.3), to estimate a median access time of operations *EdgeExists* and *Successors* in our experimental setup. This median access time combines the access times of operations *EdgeExists* and *Successors* by considering the probability of an edge existing either in the compressed diagonal stripe or in the rest of the graph, and is represented by BV+ (total) in Table 5.5. For the *Successors* query, the complete list of successors can be reported after both operations (BV+ (D) and BV+ (Non-D)) are executed. Executing these operations in parallel limits the required time of our method to at most that of the lengthier one, i.e., BV+ (Non-D)), which is less than the time BV needs. However, by populating the list of successors with part of the result right after the faster operation finishes (using the Java *BlockingQueue* interface), we enable the user to utilize it earlier. Thus, the overall execution time is approximated by BV+ (total). Of course, we also present the average time these queries needed when the full graph is compressed using the BV method.

The tests were applied to a web (`cnr-2000`), a road network (`roadNet-CA`), and a social network (`ljournal-2008`) graph. The *k* and *b* parameters of BV+ were the ones used in Table 5.3. We see that BV+ can answer both queries in constant time as far as the edges in the compressed diagonal are concerned. This time is much smaller than the time needed for the BV method to answer for the rest of the edges. In addition to this, we notice that for all graphs the BV method benefits from having to compress less edges. Unsurprisingly, the time needed to answer the two queries is larger when all the edges of the graph are compressed with the BV method. The BV+ method needs to address queries for both cases (edge lies inside or outside the compressed diagonal), but this does not impose an additional overhead to any non-single core environment, as the tasks are clearly separated. Thus, BV+ outperforms BV as far as access times are concerned for all the graphs tested. The better access times for edges belonging either to the diagonal stripe or to the rest of the graph reflect to BV+ (total). In particular, operations *EdgeExists* and *Successors* run faster with BV+ than with BV by 8.75% and 13.48% for graph `cnr-2000`, by 21.13% and 13.80% for graph `roadNet-CA`, and by 9.50% and 8.41% for graph `ljournal-2008` respectively.

As is the case with the compression ratio comparison that took place in Section 5.3.3.2, BV+ outperforms BV regarding access times too.

5.3.3.4 The effect of $\mathrm{BV}+$ parameters

The results illustrated in Table 5.3 highlight among other things the important role parameters k and b play in obtaining a good compression ratio for a given graph. We remind the reader that parameter k determines the width of the diagonal stripe of a

graph; the width is equal to $2k + 1$. For example, a 3-diagonal stripe of a particular graph is illustrated in Figure 5.8. Parameter b denotes the number of bits that comprise a row of the compressed diagonal stripe. In particular, b-digit binary numbers are used to represent the $2^b - 1$ numbers that are met most frequently among the rows of the diagonal stripe. Using Proposition 6, we can estimate how much better than the state-of-the-art-method we can represent the dense part of the graph for a given k, but the pair that produces the optimal result is highly dependent on the structure of the graph and parameter b, since there is a trade-off between keeping the ratio of the compressed diagonal stripe low and including as many edges as possible in it. For example, for the graph `roadNet-PA`, the best pair turned out to be $\{k = 7, b = 6\}$ which gives a ratio of 3.27 bits/edge, which is worse than the one given for $\{k = 7, b = 2\}$ (1.95 bits/edge), but includes almost twice as many edges.

The values of parameters k and b are fixed by performing a statical analysis per given graph prior to its compression. For the dataset that we have experimented with, we have found that a good selection of values for parameter k ranges from 2 to 20 for k, and b should be at most equal to k. However, for the sake of presenting the best possible results in this chapter, we even went further and tested values that were outside the aforementioned ranges, to come up with a pair that results in the best compression ratio for each graph.

We can see that for graphs `dblp-2010` and `amazon-2008` the selected value of parameter k is outside the range $[2, 20]$, thus, seemingly unsettling our initial argument about having come up with a proper range of k. However, the fact is that we had obtained a very good compression ratio, very close to the one presented in Table 5.3, with a value of k between 2 and 20 for both of these graphs. In particular, for `dblp-2010` we achieved a compression ratio of 7.23 for $k = 16$ and $b = 6$, which is only slightly worse to the compression ratio of 7.20 for $k = 24$ and $b = 7$. And for `amazon-2008`, we achieved a compression ratio of 10.078 for $k = 20$ and $b = 15$, which is almost identical to the compression ratio of 10.074 for $k = 23$ and $b = 15$.

In any case, we chose to use the values of parameters k and b that provided us with the best compression ratio, since we felt that it was very important for us to present the best results obtained in the experimental evaluation of our method. This goes to say that we have identified a strong trend for the values of parameters k and b, but not a pattern, as there are times that our statical analysis proposes values out of the aforementioned range. The important fact to note is that by thoroughly testing several graphs for various values of k we observed that the compression using our algorithm, viz., BV+, is better than the compression using algorithm BV for all of the graphs tested.

5.4 Discussion and Open Questions

In this chapter we propose a simple method for exploiting a particular property of social network graphs, namely, locality, in a more effective way than the state-of-the-art method of Boldi et al. [29, 30]. Our experiments point out that our method achieves higher compression rates on a broad dataset of social network graphs, while also offering constant retrieval time for the diagonal part of the graph.

An open direction is the issue of optimizing the representation of the diagonal stripe by further decreasing the total compression ratio, preferably without introducing a significant access time overhead. Moreover, our intuition suggests that a rigorous study of graph reordering methods will lead to the identification of even more attractive labellings for our proposal.

In [95] we went beyond the state-of-the-art method of Boldi and Vigna for the compression of web graphs [29] by exploiting the clustering properties observed in graphs that represent networks created by human activity, like the worldwide web or social networks, in a way different than in [29]. Essentially, we modified the way [29] represents a dense subgraph of such graphs, by exploiting their properties, namely, *locality of reference* and *similarity*. Experimental evaluation of our approach on a wide and carefully selected dataset of graphs shows remarkable decrease of the graphs' compressed size, that reaches up to 16.6%. Moreover our approach provides up to 21.13% faster access on the graphs' elements. As is the case with *BV* [29], we can also get even better results by applying *BV+* after having reordered the graph, using certain reordering algorithms, with the one presented in [30] being an obvious example.

In the scope of this work, we thoroughly investigated the research activity on compressed data structures for graphs, and presented here the most eminent of those approaches, which managed to introduce significant advances in the field of graph compression.

Our work leaves various interesting aspects open. Choosing an appropriate set for representing values of the diagonal stripe may become even more effective by using a heuristic different than promoting the ones with high frequency and a significant amount of edges, thus, leading to a more efficient compression of the graph.

Furthermore, it seems that compressing the diagonal stripe in a way different than the rest graph improves the overall compression, at least when building on the method of Boldi and Vigna as we have shown in this chapter. We need to specify the reason why this happens, and possibly explore this question for other implementations as well.

Aside the compression issues, we observed that in the visual representations of the adjacency matrices corresponding to the *dblp-2010* and *amazon-2008* graphs shown

in Figure 5.9, there is a large number of edges forming a structure that resembles a parabolic curve on the upper left hand side of each matrix. We would like to have an intuitive explanation of this structure.

Part IV

Conclusion

Chapter 6

Conclusions and Open Directions

6.1 Conclusions

In the first part of this thesis we presented two frameworks for the study of selfish user behavior in routing and information networks respectively. Concerning selfish routing, we studied games on parallel links and considered symmetric players. We saw that these games possess a unique Nash equilibrium, in which the probabilities drop linearly with time on each link. In the optimal setting of the system the probabilities are of the same form, but the players assign lower probability at the beginning of the game, so the probabilities span to more strategies. The price of anarchy of this game is 1.06.

In the second part we proposed method for compressing networks created by human activity, that outperforms the current state-of-the-art method, and provided analysis and experimental evaluation of the method. Our method can be combined with any graph compression algorithm. We employed the algorithm BV and compressed the graph after reordering its nodes using LLP algorithm.

6.2 Open Directions

This thesis consists of a few steps towards addressing the problems discussed in the previous parts, but leaves many interesting problems open. We point out here some of these directions.

It would be interesting to consider more general cases of the routing games we presented, for example non-symmetric players, more general network structures, or more than two players in the conveyor belt game.

Regarding the study of the worldwide web, it would be interesting to study the effect of link prices on PageRank and design mechanisms that could improve the efficiency of web search.

Our graph compression method could be improved if we employed more favorable

labelings. However, finding the labeling that minimizes some objective function is an NP-hard problem, so this direction is challenging. Moreover, we would be interested to see graph compression methods that allow efficient retrieval not only for the outgoing neighbors of a specific node, but also for the incoming ones.

Abbreviations - Acronyms

BV	Graph compression framework of Boldi and Vigna ([29])
BV+	Graph compression method of Liakos, Papakonstantinopoulou and Sioutis ([95])
LLP	the *Layered Label Propagation* algorithm of Boldi et al. ([30])

References

[1] Lada A Adamic and Bernardo A Huberman. Power-law distribution of the world wide web. *Science*, 287(5461):2115–2115, 2000.

[2] Micah Adler and Michael Mitzenmacher. Towards Compressing Web Graphs. In *DCC*, 2001.

[3] William Aiello, Fan Chung, and Linyuan Lu. Random evolution in massive graphs. In James Abello, Panos M. Pardalos, and Mauricio G. C. Resende, editors, *Handbook of massive data sets*, pages 97–122. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[4] A. Akella, S. Seshan, R. Karp, S. Shenker, and C. Papadimitriou. Selfish behavior and stability of the internet: a game-theoretic analysis of TCP. In *Proceedings of the 2002 SIGCOMM conference*, pages 117–130. ACM, 2002.

[5] Susanne Albers, Stefan Eilts, Eyal Even-Dar, Yishay Mansour, and Liam Roditty. On nash equilibria for a network creation game. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, SODA '06, pages 89–98, New York, NY, USA, 2006. ACM.

[6] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Internet: Diameter of the world-wide web. *Nature*, 401(6749):130–131, 1999.

[7] Eitan Altman, Rachid El Azouzi, and Tania Jiménez. Slotted aloha as a game with partial information. *Comput. Netw.*, 45:701–713, August 2004.

[8] Eitan Altman, Dhiman Barman, Rachid El Azouzi, and Tania Jiménez. A game theoretic approach for delay minimization in slotted ALOHA. In *IEEE International Conference on Communications*, 2004.

[9] Reid Andersen, Christian Borgs, Jennifer Chayes, John Hopcroft, Vahab S. Mirrokni, and Shang-Hua Teng. Local computation of pagerank contributions. In *Proceedings of the 5th international conference on Algorithms and models for the web-graph*, WAW'07, pages 150–165, Berlin, Heidelberg, 2007. Springer-Verlag.

[10] Elliot Anshelevich, Anirban Dasgupta, Jon M. Kleinberg, Éva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. In *Proceedings of FOCS '04*, pages 295–304, 2004.

[11] Elliot Anshelevich, Anirban Dasgupta, Eva Tardos, and Tom Wexler. Near-optimal network design with selfish agents. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, STOC '03, pages 511–520, New York, NY, USA, 2003. ACM.

[12] Elliot Anshelevich and Satish Ukkusuri. Equilibria in dynamic selfish routing. In *Proceedings of SAGT '09*, pages 171–182, Berlin, Heidelberg, 2009. Springer-Verlag.

[13] Alberto Apostolico and Guido Drovandi. Graph Compression by BFS. *Algorithms*, 2(3):1031–1044, 2009.

[14] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[15] Yasuhito Asano, Yuya Miyawaki, and Takao Nishizeki. Efficient Compression of Web Graphs. In *COCOON*, 2008.

[16] David Avis, Kazuo Iwama, and Daichi Paku. Verifying nash equilibria in pagerank games on undirected web graphs. In *Proceedings of the 22nd international conference on Algorithms and Computation*, ISAAC'11, pages 415–424, Berlin, Heidelberg, 2011. Springer-Verlag.

[17] David Avis, Kazuo Iwama, and Daichi Paku. Reputation games for undirected graphs. *CoRR*, abs/1205.6683, 2012.

[18] Konstantin Avrachenkov and Dmitri Lebedev. Pagerank of scale-free growing networks. *Internet Mathematics*, 3(2):207–231, 2007.

[19] Konstantin Avrachenkov and Nelly Litvak. The effect of new links on google pagerank. *Stoch. Models*, 22:2006, 2006.

[20] Baruch Awerbuch, Yossi Azar, and Amir Epstein. Large the price of routing unsplittable flow. In *Proceedings of STOC '05*, pages 57–66, 2005.

[21] Ricardo Baeza-Yates, Carlos Castillo, and Vicente López. Pagerank increase under different collusion topologies. In *First International Workshop on Adversarial Information Retrieval on the Web*, pages 17–24, May 2005.

[22] Bahman Bahmani, Ravi Kumar, Mohammad Mahdian, and Eli Upfal. Pagerank on an evolving graph. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 24–32, New York, NY, USA, 2012. ACM.

[23] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[24] Albert-László Barabási, Réka Albert, and Hawoong Jeong. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, 281(1):69–77, 2000.

[25] Nadine Baumann and Sebastian Stiller. The price of anarchy of a network creation game with exponential payoff. In Burkhard Monien and Ulf-Peter Schroeder, editors, *Algorithmic Game Theory, First International Symposium, SAGT 2008, Paderborn, Germany, April 30-May 2, 2008. Proceedings*, volume 4997 of *Lecture Notes in Computer Science*, pages 218–229. Springer, 2008.

[26] Luca Becchetti and Carlos Castillo. The distribution of pagerank follows a power-law only for particular values of the damping factor. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 941–942, New York, NY, USA, 2006. ACM.

[27] Krishna Bharat, Andrei Broder, Monika Henzinger, Puneet Kumar, and Suresh Venkatasubramanian. The connectivity server: fast access to linkage information on the Web. In *WWW*, 1998.

[28] Umang Bhaskar, Lisa Fleischer, and Elliot Anshelevich. A stackelberg strategy for routing flow over time. In *Proceedings of SODA '11*, pages 192–201. SIAM, 2011.

[29] P. Boldi and S. Vigna. The WebGraph Framework I: Compression Techniques. In *WWW*, 2004.

[30] Paolo Boldi, Marco Rosa, Massimo Santini, and Sebastiano Vigna. Layered Label Propagation: A MultiResolution Coordinate-Free Ordering for Compressing Social Networks. In *WWW*, 2011.

[31] Paolo Boldi, Massimo Santini, and Sebastiano Vigna. Permuting Web and Social Graphs. *Internet Mathematics*, 6(3):257–283, 2009.

[32] Béla Bollobás, Christian Borgs, Jennifer Chayes, and Oliver Riordan. Directed scale-free graphs. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '03, pages 132–139, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.

[33] Béla Bollobás, Oliver Riordan, Joel Spencer, and Gábor Tusnády. The degree sequence of a scale-free random graph process. *Random Structures Algorithms*, 18(3):279–290, May 2001.

[34] Christian Borgs, Michael Brautbar, Jennifer T. Chayes, and Shang-Hua Teng. A sublinear time algorithm for pagerank computations. In Anthony Bonato and Jeannette C. M. Janssen, editors, *WAW*, volume 7323 of *Lecture Notes in Computer Science*, pages 41–53. Springer, 2012.

[35] Christian Borgs, Jennifer Chayes, Constantinos Daskalakis, and Sebastien Roch. First to market is not everything: an analysis of preferential attachment with fitness. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, STOC '07, pages 135–144, New York, NY, USA, 2007. ACM.

[36] Ulrik Brandes, Martin Hoefer, and Bobo Nick. Network creation games with disconnected equilibria. In *Proceedings of the 4th International Workshop on Internet and Network Economics*, WINE '08, pages 394–401, Berlin, Heidelberg, 2008. Springer-Verlag.

[37] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, WWW7, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[38] Nieves R. Brisaboa, Susana Ladra, and Gonzalo Navarro. k2-Trees for Compact Web Graph Representation. In *SPIRE*, 2009.

[39] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, pages 309–320, Amsterdam, The Netherlands, The Netherlands, 2000. North-Holland Publishing Co.

[40] Andrei Z. Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet L. Wiener. Graph structure in the Web. *Computer Networks*, 33(1-6):309–320, 2000.

[41] Pierce G. Buckley and Deryk Osthus. Popularity based random graph models leading to a scale-free degree sequence. *Discrete Mathematics*, 282:53–68, 2001.

[42] Gregory Buehrer and Kumar Chellapilla. A scalable pattern mining approach to web graph compression with communities. In *WSDM*, 2008.

[43] Wei Chen, Shang-Hua Teng, Yajun Wang, and Yuan Zhou. On the alpha-sensitivity of nash equilibria in pagerank-based network reputation games. In *Proceedings of the 3d International Workshop on Frontiers in Algorithmics*, FAW '09, pages 63–73, Berlin, Heidelberg, 2009. Springer-Verlag.

[44] Yen-Yu Chen, Qingqing Gan, and Torsten Suel. Local methods for estimating pagerank values. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 381–389, New York, NY, USA, 2004. ACM.

[45] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, Michael Mitzenmacher, Alessandro Panconesi, and Prabhakar Raghavan. On compressing social networks. In *KDD*, 2009.

[46] G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of STOC '05*, pages 67–73. ACM, 2005.

[47] G. Christodoulou, E. Koutsoupias, and P. Spirakis. On the performance of approximate equilibria in congestion games. In *European Symposium on Algorithms*, pages 251–262, Copenhagen, Denmark, 7-9 September 2009. Also in arXiv:cs/0804.3160.

[48] George Christodoulou, Elias Koutsoupias, and Akash Nanavati. Coordination mechanisms. *Theor. Comput. Sci.*, 410(36):3327–3336, 2009.

[49] George Christodoulou, Katrina Ligett, and Evangelia Pyrga. Contention resolution under selfishness. In *Proceedings of ICALP 2010*, pages 430–441. Springer-Verlag.

[50] Francisco Claude and Susana Ladra. Practical Representations for Web and Social Graphs. In *CIKM*, 2011.

[51] Francisco Claude and Gonzalo Navarro. A Fast and Compact Web Graph Representation. In *SPIRE*, 2007.

[52] Colin Cooper and Alan Frieze. A general model of web graphs. *Random Struct. Algorithms*, 22(3):311–335, May 2003.

[53] Andreas Cord-Landwehr, Martina Hüllmann, Peter Kling, and Alexander Setzer. Basic network creation games with communication interests. In *Proceedings of the 5th international conference on Algorithmic Game Theory*, SAGT'12, pages 72–83, Berlin, Heidelberg, 2012. Springer-Verlag.

[54] Mark E Crovella and Azer Bestavros. Explaining world wide web traffic self-similarity. Technical report, Boston University Computer Science Department, 1995.

[55] Mark E Crovella and Azer Bestavros. Self-similarity in world wide web traffic: evidence and possible causes. *Networking, IEEE/ACM Transactions on*, 5(6):835–846, 1997.

[56] Mark E Crovella, Murad S Taqqu, and Azer Bestavros. Heavy-tailed probability distributions in the world wide web. *A practical guide to heavy tails*, 1:3–26, 1998.

[57] Balázs Csanád Csáji, Raphaël M. Jungers, and Vincent D. Blondel. Pagerank optimization in polynomial time by stochastic shortest path reformulation. In *Proceedings of the 21st international conference on Algorithmic learning theory*, ALT'10, pages 89–103, Berlin, Heidelberg, 2010. Springer-Verlag.

[58] Charo I. Del Genio, Thilo Gross, and Kevin E. Bassler. All Scale-Free Networks Are Sparse. *Phys. Rev. Lett.*, 107:178701, 2011.

[59] Erik D. Demaine, Mohammad Taghi Hajiaghayi, Hamid Mahini, and Morteza Zadimoghaddam. The price of anarchy in cooperative network creation games. *SIGecom Exchanges*, 8(2):2, 2009.

[60] Erik D. Demaine, Mohammad Taghi Hajiaghayi, Hamid Mahini, and Morteza Zadimoghaddam. The price of anarchy in network creation games. *ACM Transactions on Algorithms*, 8(2):13, 2012.

[61] Erik D. Demaine and Morteza Zadimoghaddam. Constant price of anarchy in network creation games via public service advertising. In *Algorithms and models for the web graph. 7th international workshop, WAW 2010, Stanford, CA, USA, December 16, 2010. Proceedings*, pages 122–131. Berlin: Springer, 2010.

[62] S N Dorogovtsev, J FF Mendes, and A N Samukhin. Structure of growing networks with preferential linking. *Physical Review Letters*, 85,:4633–4636, 2000.

[63] Eleni Drinea, Mihaela Enachescu, and Michael Mitzenmacher. Variations on random graph models for the web. Technical report, Harvard University, 2001.

[64] Éva Tardos and Tom Wexler. Network formation games and the potential function method. In *Algorithmic Game Theory, chapter 19*, pages 487–516, 2007.

[65] Alex Fabrikant, Ankur Luthra, Elitza Maneva, Christos H. Papadimitriou, and Scott Shenker. On a network creation game. In *Proceedings of the twenty-second annual symposium on Principles of distributed computing*, PODC '03, pages 347–351, New York, NY, USA, 2003. ACM.

[66] Amos Fiat, Yishay Mansour, and Uri Nadav. Efficient contention resolution protocols for selfish agents. In *Proceedings of SODA '07*, pages 179–188, 2007.

[67] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[68] R. Garg, A. Kamra, and V. Khurana. A game-theoretic approach towards congestion control in communication networks. *ACM SIGCOMM Computer Communication Review*, 32(3):47–61, 2002.

[69] P. Giammatteo, D. Donato, V. Zlatić, and G. Caldarelli. A pagerank-based preferential attachment model for the evolution of the world wide web. *EPL (Europhysics Letters)*, 91(1):18004, 2010.

[70] Cecilia Hernández and Gonzalo Navarro. Compressed representation of web and social networks via dense subgraphs. In *SPIRE*, 2012.

[71] Martin Hoefer, Vahab S. Mirrokni, Heiko Röglin, and Shang-Hua Teng. Competitive routing over time. In *Proceedings of WINE 2009*, pages 18–29. Springer-Verlag.

[72] John Hopcroft and Daniel Sheldon. Manipulation-resistant reputations using hitting time. In *Proceedings of the 5th international conference on Algorithms and models for the web-graph*, WAW'07, pages 68–81, Berlin, Heidelberg, 2007. Springer-Verlag.

[73] John Hopcroft and Daniel Sheldon. Network reputation games, 2008.

[74] Bernardo A Huberman and Lada A Adamic. Internet: growth dynamics of the world-wide web. *Nature*, 401(6749):131–131, 1999.

[75] Bernardo A. Huberman and Lada A. Adamic. The nature of markets in the world wide web. Computing in Economics and Finance 1999 521, Society for Computational Economics, March 1999.

[76] Bernardo A Huberman, Peter LT Pirolli, James E Pitkow, and Rajan M Lukose. Strong regularities in world wide web surfing. *Science*, 280(5360):95–97, 1998.

[77] Nicole Immorlica, Kamal Jain, and Mohammad Mahdian. Game-theoretic aspects of designing hyperlink structures. In *Internet and Network Economics, Second International Workshop, WINE 2006*, pages 150–161, 2006.

[78] Matthew O. Jackson. A survey of models of network formation: Stability and efficiency," mimeo. In *California Institute of Technology*, 2003.

[79] U. Kang and Christos Faloutsos. Beyond “caveman communities”: Hubs and spokes for graph compression and mining. In *ICDM*, 2011.

[80] Zsolt Katona and Miklos Sarvary. Network formation and the structure of the commercial world wide web. *Marketing Science*, 27(5):764–778, 09-10 2008.

[81] Cristobald Kerchove de, Laure Ninove, and Paul Dooren van. Maximizing pagerank via outlinks. *Linear Algebra and its Applications*, 429:1254–1276, 2008.

[82] A. Kesselman, S. Leonardi, and V. Bonifaci. Game-theoretic analysis of internet switching with selfish users. *Internet and Network Economics*, pages 236–245, 2005.

[83] Jon Kleinberg. The small-world phenomenon: an algorithm perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 163–170. ACM, 2000.

[84] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September 1999.

[85] Jon M Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S Tomkins. The web as a graph: Measurements, models, and methods. In *Computing and combinatorics*, pages 1–17. Springer, 1999.

[86] Lasse Kliemann. The price of anarchy for network formation in an adversary model. *Games*, 2(3):302–332, 2011.

[87] Ronald Koch and Martin Skutella. Nash equilibria and the price of anarchy for flows over time. In *Proceedings of SAGT ’09*, pages 323–334. Springer-Verlag, 2009.

[88] Scott Duke Kominers. Sticky content and the structure of the commercial web. Technical report, Harvard University, 2009.

[89] George Kouroupas, Elias Koutsoupias, Christos H. Papadimitriou, and Martha Sideri. An economic model of the worldwide web. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, WWW '05, pages 934–935, New York, NY, USA, 2005. ACM.

[90] Georgios Kouroupas, Elias Koutsoupias, Christos H. Papadimitriou, and Martha Sideri. Experiments with an economic model of the worldwide web. In *Proceedings of the First international conference on Internet and Network Economics*, WINE'05, pages 46–54, Berlin, Heidelberg, 2005. Springer-Verlag.

[91] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *Proceedings of STACS '99*, pages 404–413, Berlin, Heidelberg, 1999. Springer-Verlag.

[92] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *WWW*, 2010.

[93] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations. In *KDD*, 2005.

[94] Panagiotis Liakos, Katia Papakonstantinopoulou, and Michael Sioutis. On the Effect of Locality in Compressing Social Networks. In *ECIR*, 2014.

[95] Panagiotis Liakos, Katia Papakonstantinopoulou, and Michael Sioutis. Pushing the envelope in graph compression. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, CIKM '14, pages 1549–1558, New York, NY, USA, 2014. ACM.

[96] Allen B. MacKenzie and Stephen B. Wicker. Stability of multipacket slotted aloha with selfish users and perfect information. In *Proceedings of IEEE INFOCOM*, pages 1583–1590, 2003.

[97] Martin Macko, Kate Larson, and L'uboš Steskal. Braess's paradox for flows over time. In *Proceedings of SAGT '10*, pages 262–275. Springer-Verlag, 2010.

[98] Hossein Maserrat and Jian Pei. Neighbor query friendly compression of social networks. In *KDD*, 2010.

[99] Iain Middleton, Mike McConnell, and Grant Davidson. Presenting a model for the structure and content of a university world wide web site. *Journal of Information Science*, 25(3):219–227, 1999.

[100] Matúš Mihalák and Jan Christoph Schlegel. The price of anarchy in network creation games is (mostly) constant. In *Proceedings of the Third international conference on Algorithmic game theory*, SAGT'10, pages 276–287, Berlin, Heidelberg, 2010. Springer-Verlag.

[101] D. Monderer and L.S. Shapley. Potential games. *Games and economic behavior*, 14:124–143, 1996.

[102] John Nash. Non-cooperative Games. *The Annals of Mathematics*, 54(2):286–295, 1951.

[103] N. Nisan, T. Roughgarden, É. Tardos, and V.V. Vazirani. *Algorithmic game theory*. Cambridge Univ Pr, 2007.

[104] Martin Olsen. The computational complexity of link building. In *Computing and Combinatorics, 14th Annual International Conference*, pages 119–129, 2008.

[105] Martin Olsen. Maximizing pagerank with new backlinks. In *CIAC*, pages 37–48, 2010.

[106] Martin Olsen and Anastasios Viglas. On the approximability of the link building problem. *Theoretical Computer Science*, 518(0):96 – 116, 2014.

[107] Martin Olsen, Anastasios Viglas, and Ilia Zvedeniouk. A constant-factor approximation algorithm for the link building problem. In Weili Wu and Ovidiu Daescu, editors, *COCOA (2)*, volume 6509 of *Lecture Notes in Computer Science*, pages 87–96. Springer, 2010.

[108] M.J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[109] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[110] Gopal Pandurangan, Prabhakar Raghavan, and Eli Upfal. Using pagerank to characterize web structure. In *Proceedings of the 8th Annual International Conference on Computing and Combinatorics*, COCOON '02, pages 330–339, London, UK, UK, 2002. Springer-Verlag.

[111] Sriram Raghavan and Hector Garcia-Molina. Representing Web Graphs. In *ICDE*, 2003.

[112] Keith H. Randall, Raymie Stata, Janet L. Wiener, and Rajiv G. Wickremesinghe. The Link Database: Fast Access to Graphs of the Web. In *DCC*, 2002.

[113] R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[114] Tim Roughgarden. Intrinsic robustness of the price of anarchy. In *Proceedings of STOC '09*, pages 513–522, New York, NY, USA, 2009. ACM.

[115] Tim Roughgarden and Éva Tardos. How bad is selfish routing? *J. ACM*, 49(2):236–259, March 2002.

[116] Ilya Safro and Boris Temkin. Multiscale approach for the network compression-friendly ordering. *J. of Discrete Algorithms*, 9(2):190–202, 2011.

[117] Piotr Sankowski. Dynamic transitive closure via dynamic matrix inverse (extended abstract). In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '04, pages 509–517, Washington, DC, USA, 2004. IEEE Computer Society.

[118] Claude Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423, 1948.

[119] Jirí Síma and Satu Elisa Schaeffer. On the NP-Completeness of Some Graph Cluster Measures. In *SOFSEM*, 2006.

[120] Torsten Suel and Jun Yuan. Compressing the Graph Structure of the Web. In *DCC*, 2001.

[121] Bosiljka Tadić. Dynamics of directed graphs: the world-wide web. *Physica A: Statistical Mechanics and its Applications*, 293(1):273–284, 2001.

[122] Yana Volkovich, Nelly Litvak, and Debora Donato. Determining factors behind the pagerank log-log plot. In *Proceedings of the 5th international conference on Algorithms and models for the web-graph*, WAW'07, pages 108–123, Berlin, Heidelberg, 2007. Springer-Verlag.

[123] Ian H. Witten, Timothy C. Bell, and Alistair Moffat. *Managing Gigabytes: Compressing and Indexing Documents and Images*. John Wiley & Sons, Inc., 1st edition, 1994.

[124] Yi Zhang, Kaihua Xu, Yuhua Liu, and Zhenrong Luo. Modeling of scale-free network based on pagerank algorithm. In *Future Computer and Communication*

(ICFCC), 2010 2nd International Conference on, volume 3, pages V3–783–V3–786, 2010.