ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΛΟΓΙΚΗΣ ΚΑΙ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΜΩΝ (Co.Re.Lab.)

Παίγνια Συμφόρησης: Στοχαστικές Επεκτάσεις και Τεχνικές Μείωσης του Τιμήματος της Αναρχίας

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

του

Αθανασίου Β. Λιανέα

Αθήνα, Δεκέμβριος 2014

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΛΟΓΙΚΗΣ ΚΑΙ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΜΩΝ

Παίγνια Συμφόρησης: Στοχαστικές Επεκτάσεις και Τεχνικές Μείωσης του Τιμήματος της Αναρχίας

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

του

Αθανασίου Β. Λιανέα

Συμβουλευτική Επιτροπή: Ευστάθιος Ζάχος (Επιβλέπων)
Δημήτριος Φωτάκης
Αριστείδης Παγουρτζής

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 16η Δεκεμβρίου 2014.

.....	.....	.....
Ευστάθιος Ζάχος Καθηγητής ΕΜΠ	Δημήτριος Φωτάκης Επ. Καθηγητής ΕΜΠ	Αριστείδης Παγουρτζής Επ. Καθηγητής ΕΜΠ
.....	.....	.....
Αντώνιος Συνμβώνης Καθηγητής ΕΜΠ	Σταύρος Κολλιόπουλος Αν. Καθηγητής ΕΚΠΑ	Ευάγγελος Μαρκάκης Επ. Καθηγητής ΟΠΑ

.....
Βασίλειος Ζησιμόπουλος
Καθηγητής ΕΚΠΑ

Αθήνα, Δεκέμβριος 2014.

.....
Αθανάσιος Β. Λιανέας
Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Το αντικείμενο της διατριβής είναι η θεωρητική ανάλυση και γενίκευση μοντέλων παιγνίων συμφόρησης, με στόχο την μελέτη μεθόδων μείωσης του Τιμήματος της Αναρχίας και τη μελέτη στοχαστικών επεκτάσεων των παιγνίων συμφόρησης με παράλληλη έρευνα του κατά πόσο μπορούν αυτές να επηρεάσουν, είτε θετικά είτε αρνητικά, το Τίμημα της Αναρχίας. Αρχικά, παρουσιάζονται βασικά στοιχεία της βιβλιογραφίας που έχουν άμεση σχέση με τα προβλήματα που μελετήθηκαν και στην συνέχεια ακολουθεί μια εκτεταμένη παρουσίαση των αποτελεσμάτων της εργασίας.

Παρουσιάζονται αποτελέσματα που αφορούν το παράδοξο του Braess σε παίγνια συμφόρησης όπου το κόστος κάθε παίκτη ισούται με το κόστος της πιο ακριβής ακμής (ακμή συμφόρησης) που χρησιμοποιεί και το κόστος του δικτύου ισούται με το κόστος της πιο ακριβής ακμής που χρησιμοποιείται. Μελετάται το πρόβλημα εύρεσης έστω και προσεγγιστικά καλύτερου υποδικτύου σε τέτοιου είδους παίγνια. Παρότι το αντίστοιχο πρόβλημα σε παίγνια με προσθετικά κόστη είχε κατηγοριοποιηθεί από πλευράς χρονικής πολυπλοκότητας για το εν λόγω πρόβλημα υπήρχαν κάποια αποτελέσματα μόνο για γενικεύσεις του και μάλιστα αρκετά πιο ασθενή από αυτά που παρουσιάζονται στην διατριβή. Για την πιο απλή εκδοχή του προβλήματος, μέσω μιας σύνθετης αναγωγής, αποδεικνύονται αποτελέσματα δυσκολίας στην προσέγγιση του καλύτερου υποδικτύου: είναι NP-δύσκολο το πρόβλημα εύρεσης έστω και $O(n^{0.121})$ προσεγγιστικά καλού υποδικτύου (όπου n ο αριθμός κόμβων δικτύου). Παράλληλα καταδεικνύονται δύο υποκατηγορίες τέτοιων παιγνίων που δεν πάσχουν καθόλου από το παράδοξο ενώ δίνεται ένας προσεγγιστικός αλγόριθμος για περιπτώσεις δικτύων όπου η αναγωγή της απόδειξης της NP-δυσκολίας δεν μπορεί να εφαρμοστεί.

Παρουσιάζονται, επίσης, αποτελέσματα που έχουν να κάνουν με το παράδοξο του Braess σε παίγνια συμφόρησης με προσθετικά κόστη. Για τέτοιου είδους παίγνια επάνω σε Erdös-Rényi τυχαίους γράφους (ως υποκείμενα δίκτυα) έχει αποδειχτεί ότι με μεγάλη πιθανότητα εμφανίζεται το παράδοξο του Braess. Στην εργασία αυτή, το πρόβλημα εύρεσης καλύτερου υποδικτύου σε τέτοιου είδους παίγνια, διαισθητικά ανάγεται σε πρόβλημα εύρεσης καλύτερου υποδικτύου σε παίγνια όπου το υποκείμενο δίκτυο ανήκει στην πιο απλή οικογένεια υποδικτύων που θα μπορούσαν να πάσχουν από το παράδοξο, με την δυσκολία εύρεσης καλύτερου υποδικτύου σε τέτοια δίκτυα να παραμένει άγνωστη. Χρησιμοποιώντας ένα πολύ πρόσφατο αποτέλεσμα από τη θεωρία

πιθανοτήτων, δίνεται ένας πολυωνυμικός αλγόριθμος προσέγγισης του καλύτερου υποδικτύου σε τέτοια δίκτυα και ακολούθως, χρησιμοποιώντας τις επεκτατικές ιδιότητες των Erdös-Rényi γράφων, άγεται ένα προσεγγιστικά καλό υποδίκτυο του αρχικού δικτύου.

Σε λιγο διαφορετική κατεύθυνση, μελετώνται τα βασικά χαρακτηριστικά παιγνίων συμφόρησης με αβεβαιότητα στις ακμές και παίκτες ευαίσθητους στο ρίσκο. Η κλασική μοντελοποίηση των παιγνίων συμφόρησης αγνοεί την αβεβαιότητα στις ακμές που ενυπάρχει σε αρκετές περιπτώσεις της καθημερινότητας. Μοντελοποιώντας την αιτία της αβεβαιότητας στα κόστη των ακμών, στην εργασία, ορίζονται δύο ¨ορθογώνια¨ μοντέλα, ένα με στοχαστικούς παίκτες, όπου οι παίκτες συμμετέχουν ή όχι στο παίγνιο με δεδομένη πιθανότητα και άρα η πραγματική συμφόρηση για τις ακμές που επιλέγουν αποκτά τυχαιότητα, και ένα με στοχαστικές ακμές, όπου οι ακμές δύνανται με κάποια πιθανότητα να έχουν ¨μη-κανονική¨ συμπεριφορά και να προσδίδουν μεγαλύτερη καθυστέρηση κατά την χρήση τους. Σε αυτά τα παίγνια γίνεται μελέτη ως προς την ύπαρξη σημείων ισορροπίας και συναρτήσεων δυναμικού ενώ μελετάται και η συμπεριφορά του τιμήματος της αναρχίας.

Ενοποιώντας τις δύο κατευθύνσεις, στην εργασία δίδεται ένας νέος τρόπος βελτίωσης του τιμήματος της αναρχίας σε παίγνια με αβεβαιότητα στις ακμές και παίκτες ευαίσθητους στο ρίσκο. Πιο συγκεκριμένα, δείχνεται ότι αν είναι δυνατή η προσθήκη επιπλέον αβεβαιότητας σε επιλεγμένες ακμές με τρόπον ώστε να μην αλλάζει το αναμενόμενο κόστος τους, τότε, λόγω της ευαισθησίας των παικτών στο ρίσκο, αυτές οι ακμές γίνονται λιγότερο προτιμητέες από τους παίκτες και συνεπώς το τίμημα της αναρχίας δύναται να βελτιωθεί λόγω της στροφής των παικτών προς ακμές που η βέλτιστη λύση θα επέλεγε για αυτούς. Το πρόβλημα που ορίζεται προς αυτή την κατεύθυνση μοιάζει με την περιορισμένη χρήση διοδίων σε δίκτυα και αποτελέσματα μπορούν να προκύψουν από εκεί. Στην εργασία δίνονται αποτελέσματα που προσπαθώντας να ακολουθήσουν τις Karush Kuhn Tucker συνθήκες βελτιστότητας δίνουν μια ¨οικονομικότερη¨ και καλύτερη διαχείριση της αβεβαιότητας και παρέχουν καλύτερη διαίσθηση για την προκύπτουσα βελτίωση του τιμήματος της αναρχίας.

Abstract

The subject of this thesis is the theoretical analysis and generalization of congestion games models and it aims to provide a study on mehtods for reducing the Price of Anarchy and a study related to stochastic extensions of congestion games and in which extend the Price of Anarchy may be affected within them. First, the literature that relates mostly to the problems studied is presented and rightafter an extensive presentation of the results of the thesis follows.

The problem of finding or approximating the best subnetwork in bottleneck routing games is studied. Although the corresponding problem in additive costs congestion games is almost fully understood, for the problem studied here, the existing results hold for more general games and in fact are much weaker than the ones presented here. For the simplest version of this problem, via a complex reduction, an NP-hardness results for finding or approximating the best subnetwork of the underlying network is proved: it is NP-hard to approximate the best subnetwork by a factor less than $O(n^{0.121})$ (where n is the number of nodes of the network). In the positive side it is proven that in some subclasses of these games the paradox does not appear at all and also it is given an approximation algorithm for some cases where the NP-hardness reductions cannot apply.

Results that have to do with Braess paradox in additive costs congestion games are presented. Prior to the work presented here, it has been proved that if the underlying network of a congestion game is a random Erdös-Rényi graph then with high probability it suffers from Braess Paradox. Here, it is proven that the problem of finding the best subnetwork in such random networks can be essentially reduced to the problem of finding the best subnetwork of a network belonging to the simplest class of graphs that may suffer from the paradox. Using a very recent result from the theory of probabilities, it is given a polynomial approximation algorithm for finding best subnetworks in such networks. Then, the expansion properties of Erdös-Rényi graphs are used and an approximately good subnetwork for the initial network is drawn.

By slightly changing direction, the basic properties of congestion games with uncertain delays and risk averse users are studied. The classic formulation of congestion games ignores uncertainty in delays that arises in many real life situations. Modeling the cause of uncertainy in delays, two orthogonal models are introduced, one with stochastic players, where each players participates in the game with a given probability and thus the actual delay on the edges that players choose gets uncertain, and another with stochastic edges where each edge, with a given probability, may "fail" and provide greater delay to the players using it. For the arising classes of games, the existence of pure Nash equilibrium and potentials and the behavior of the price of anarchy is studied.

Uniting the above directions, a new way for improving the price of anarchy in congestion games with uncertain delays and risk averse users is given. More specifically, it is shown that if one can insert uncertainty in the edges of the network in a way that the expected cost of those edges remains the same, then, because of the risk aversion of the players, these edges become less attractive for the players. Thus, the price of anarchy may improve as more players may turn to edges that the optimal solution would choose for them. The arising algorithmic problem relates to congestion games with restricted tolls and results can also be drawn from there. In this thesis Karush Kuhn Tucker conditions are closely followed and there are given results that give a better and "less cheap" use of extra uncertainty and provide better insight of the improvement in the price of anarchy.

στους αλληλεπιδρώντες μου,
σε αντίστοιχο ποσοστό

Ευχαριστίες

Ευχαριστώ τους καθηγητές μου: Στάθη Ζάχο για την πλήρως ενεπνευστική του διδασκαλία, εντός κι εκτός αμφιθεάτρου, καθ'όλη την διάρκεια των σπουδών μου, Δημήτρη Φωτάκη που με βοήθησε υπέρ το δέον χωρίς να του ζητήσω καν βοήθεια, Άρη Παγουρτζή που με ηρεμία και καταννόηση απαντούσε και απαντά στις ερωτήσεις φοιτητών (εμού συμπεριλαμβανομένου), Νίκο Παπαγεωργίου γιατί διπλασίασε την αγάπη μου για τα μαθηματικά και Δημήτρη Πετσετίδη γιατί με εμαθε να συγκεντρώνομαι στο ζητούμενο. Επίσης οφείλω ένα μεγάλο ευχαριστώ στα υπόλοιπα μέλη της επταμελούς επιτροπής εξέτασης του παρόντος διδακτορικού, Αντώνη Συμβώνη, Σταύρο Κολλιόπουλο, Βαγγέλη Μαρκάκη και Βασίλη Ζησιμόπουλο.

Ευχαριστώ τους +αδέλφους μου στο Εργαστήριο Λογικής και Επιστήμης Υπολογισμών (Co.Re.Lab.) της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου, που βοήθησαν τόσο στην ακαδημαϊκή μου ανάπτυξη όσο και στην δημιουργία ιδανικού περιβάλλοντος στο εργαστήριο όλα αυτά τα χρόνια. Επιλέγοντας έναν αντιπρόσωπο για κάθε έτος (που όλοι στο εργαστήριο αγαπάμε όπου και αν τους βρίσκουμε), ευχαριστώ τους: Άρη Τέντε, Βαγγέλη Μπαμπά, Αντρέα Γκέμπελ, Αντρέα Γαλάνη, Θέμη Γουλεάκη, Χάρη Αγγελιδάκη, Μανώλη Ζαμπετάκη και Ναταλία Κωτσάνη. Επι τη ευκαιρία, προσθέτω ένα μεγάλο ευχαριστώ στα μέλη του πιο φοβερού study group του ΕΜΠ για όλους τους δρόμους... που βαδίσαμε μαζί.

Τέλος, ένα μεγάλο ευχαριστώ προς: τα πολυαγαπημένα μου ξαδέρφια Βασίλη, Γιωργία και Θανάση γιατί προσπαθώντας να τους μοιάσω έπρεπε να αναπτύξω πολλές πλευρές της προσωπικότητάς μου, τον αδερφό μου Γιώργο για τους ίδιους λόγους αλλά και γιατί μάλλον υπήρξε η αιτία να ανοίξω από νωρίς τα βιβλία και του γονείς μου Βασίλη και Αναστασία για την πέρα από κάθε περιγραφή βοήθειά τους όλα τα 30,997+0,75 χρόνια της ζωής μου.

Θανάσης Λιανέας
Αθήνα, Δεκέμβριος 2014

Διάρθρωση της Διατριβής

Τα παίγνια συμφόρησης αποτελούν σημαντικό κομμάτι της (αλγοριθμικής) θεωρίας παιγνίων. Αφού συνοψίσουμε κάποια βασικά στοιχεία της βιβλιογραφίας που μελετήθηκε, θα επικεντρωθούμε σε θέματα αυτής με τα οποία ασχοληθήκαμε ερευνητικά και θα παραθέσουμε τα ευρήματα των προσπαθειών μας, που έχουν να κάνουν με το παράδοξο του Braess, αλλά και με τη συμπεριφορά δύο γενικεύσεων παιγνίων συμφόρησης που προσπαθούν να προσομοιάσουν καλύτερα κάποιες πραγματικές καταστάσεις της ζωής.

Στο πρώτο κεφάλαιο συγκεντρώνουμε όλη την βιβλιογραφία που σχετίζεται με τα παίγνια συμφόρησης και την ερευνητική μας δουλειά και παρουσιάζουμε τις τεχνικές που προσπαθούν να μειώσουν την υποβάθμιση του δικτύου που προκαλεί η εγωιστική συμπεριφορά των παικτών.

Στο δεύτερο κεφάλαιο παρουσιάζουμε διαισθητικά την συνεισφορά μας ενώ στο τρίτο κεφάλαιο δίνουμε γενικούς ορισμούς που θα χρησιμοποιήσουμε στην παρουσίαση της δουλειάς μας.

Στο τέταρτο και πέμπτο κεφάλαιο θα εστιάσουμε στο παράδοξο του Braess, η εξάλειψη του οποίου αποτελεί τον πιο σίγουρο και άμεσο τρόπο για την μείωση της υποβάθμισης του δικτύου. Θα παρουσιάσουμε τεχνικά την δική μας συνεισφορά στο πεδίο αυτό με αποτελέσματα που έχουν να κάνουν 1) με την εξάλειψη του παραδόξου σε τυχαίας φύσης δίκτυα που αποδεδειγμένα πάσχουν από το παράδοξο όταν τα κόστη είναι προσθετικά και 2) με την δυσκολία εντοπισμού του παραδόξου και αποδοτικής εύρεσης έστω και προσεγγιστικά καλού υποδικτύου σε παίγνια όπου τα κόστη των μονοπατιών ισούνται με το βάρος της βαρύτερής τους ακμής.

Στο έκτο κεφάλαιο, αφού ορίσουμε δύο γενικεύσεις των παιγνίων συμφόρησης που αφορούν παίκτες ευαίσθητους στην αβεβαιότητα, θα τις αναλύσουμε ως προς την ύπαρξη ισορροπιών και συναρτήσεων δυναμικού και θα μελετήσουμε την συμπεριφορά του τιμήματος της αναρχίας, τη μονάδα μέτρησης της υποβάθμισης του δικτύου.

Στο έβδομο κεφάλαιο, δείχνουμε πως μπορούμε, σε παίγνια με παίκτες ευαίσθητους στην αβεβαιότητα, να χρησιμοποιήσουμε την ευαισθησία των παικτών και προσθέτωντας αβεβαιότητα σε μέρος του δικτύου να βελτιώσουμε την συμπεριφορά του τιμήματος της αναρχίας.

Στο όγδοο και τελευταίο κεφάλαιο, εξάγουμε τελικά συμπεράσματα, και παραθέτουμε-καταδεικνύουμε ανοικτά προβλήματα.

Περιεχόμενα

Chapter 1

Introduction

Congestion Games provide a natural model for non-cooperative resource allocation in large-scale communication networks and have been the subject of intensive research in Algorithmic Game Theory. In this chapter, we first give a small introduction to the “Congestion Games Space", inside of which all the (leading, studied) different cases and generalizations of Congestion Games (CGs in short) lie. A literature overview follows right after with emphasis given to the bibliography part that strongly relates to our work.

1.1 Congestion Games

In a *Congestion Game* [73], a set of identical non-cooperative players, controlling an equal amount of load, compete over a finite set of resources. All players using a particular resource, experience a cost (or latency) given by a non-negative and non-decreasing function of the resource’s load (or congestion). Each player selects her strategy, a subset of resources, selfishly trying to minimize her *individual cost*.

The focus on CGs is on the so-called *network* CGs where there is an underlying network given and each player’s strategy space is formed by the paths from her origin node to her destination node in the network. The edges of the network, that come together with a cost function, are the resources of the network CG. The following hold for general CGs although most of the work in the literature and also our work deals with network CGs.

Cases

Many different cases of CGs arise if one considers different assumptions on the players' strategy spaces, on the players' types or on the players' strategies' costs. More specific, players may all have the same strategy space, symmetric games case, or different strategy spaces, asymmetric games case. Also, players might be finite and route a significant amount (a unit) of load through the network, and thus may affect the resources' cost, atomic games case, or be infinite and route a negligible (infinitesimally small) amount of flow through the network and thus cannot unilaterally affect the resources' costs, non atomic games. Moreover each player's costs might be the sum of the resources' costs on player's strategy, additive costs games, or the maximum resource's cost among the resources' costs in player's strategy, bottleneck costs. Figure 1.1 schematically presents the above.

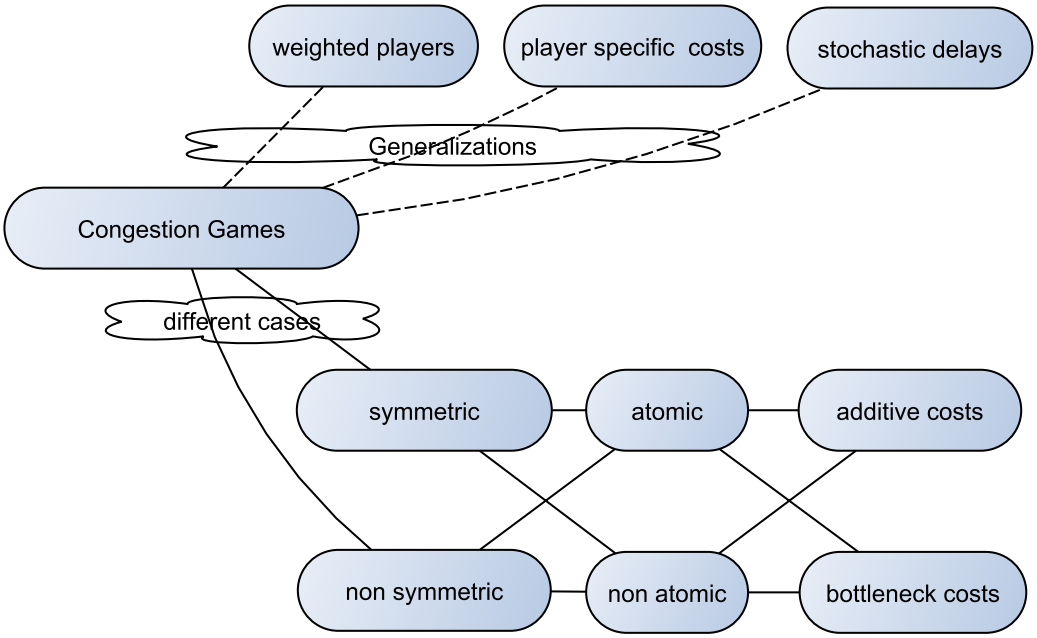

Figure 1.1: Network Congestion Games (small) map. The upper part captures the leading generalizations of CGs. The lower part captures the different cases arising by considering assumptions on the players' strategy spaces, on the players' types and on the players' strategies' costs.

Generalizations

CGs can be naturally generalized to games where players have different demands (for the atomic case), called weights, i.e. route different amounts of flow through the network (in network congestion games). These games are called *weighted* CGs. Another natural generalization assumes that resources' cost functions are player dependent, i.e. each player perceives a (possibly) different cost function on each resource. These are CGs *with player-specific cost functions*. Deciding under uncertainty has drawn much attention in CGs in the very last few years. Models of congestion games with stochastic delays and risk averse players extend CGs in order to capture real life situations and predict human behavior. Figure 1.1 helps.

Performance

In a Congestion Game, a natural solution concept is that of a *pure Nash equilibrium* (PNE), a configuration where no player can decrease her cost by unilaterally changing her strategy. At the other end, the network manager cares about the public benefit and aims to minimize the *total cost* incurred by all players. Since a Nash equilibrium does not need to optimize the total cost, one seeks to quantify the inefficiency due to selfish behavior. The *Price of Anarchy* was introduced in [57] and has become a widely accepted measure of the performance degradation due to the players' selfish behavior. The (pure) Price of Anarchy is the worst-case ratio of the total cost of a (pure) Nash equilibrium to the optimal total cost. Many recent contributions have provided strong upper and lower bounds on the Price of Anarchy (PoA) for several classes of CGs. The forthcoming literature overview includes also these results.

The prevailing questions in CGs of any of the different cases described above (or even for subcases of them), have to do with the existence and computation of equilibria, with the convergence time of better or best response dynamics (i.e. the steps to reach an equilibrium starting from an initial configuration and letting players do a better or best response, one at a time), with deriving bounds on the PoA and with exploring ways to improve network's performance (tolls, stackelberg strategies, exploiting braess paradox, see section 1.3). Citations (together with some details) of published work dealing with these questions and the main results of the literature part that relates the most to our work paper are given in the remaining of the chapter.

1.2 Bibliography Overview

Next we put together bibliography sources dealing with the prevailing questions in the different variants of CGs. The biggest portion of this literature concerns additive costs games although there is a non negligible portion concerning bottleneck costs games. For results on improving network's performance continue to section 1.3 and for games with stochastic delays see section 1.4

Additive Costs

For results on the existence of Nash equilibrium one should see [73] where the method of potential function is used to prove the existence of a pure Nash Equilibrium (PNE) in CGs and [66] where the equivalence of CGs with potential games is proved. On section 3.3 we open the first pages of these works.

For results on the complexity of finding PNE see e.g. [33], where the class PLS was proved to fully capture the difficulty of finding pure Nash equilibrium in general, symmetric or asymmetric CGs, [30], where hardness results for the existence of PNE in weighted games is given and [3] where hardness results for the existence and computation of PNE in player-specific CGs is given.

For results on the PoA in non atomic games see e.g. [74], where PoA bounds are given that actually depend only to the class of the cost functions of the resources and are independent of the network topology and [28], where a simpler proof of the previous result was given that also can be applied in more general settings. Also, one could see [64], where it is shown that topological properties might help the network in performing well.

For results on the PoA in atomic games see e.g. [8], where PoA bounds are given for cases where the costs are linear functions or polynomials of degree d, [23], where also bounds (slightly stronger for some cases) for cases where the costs are linear functions or polynomials of degree d are given but also a more general method for bounding PoA is introduced and [4], where exact bounds are given using similar techniques to [23] that also apply to weighted CGs.

For results on the convergence times to PNE see e.g. [32], where bounds on convergence time for singleton CGs (strategies are singletons) for both weighted and unweighted CGs, [39], where, among others, the convergence time to PNE is investigated for the case of network CGs with linearly independent paths. For results on the convergence times to approximate

Nash equilibrium (where players costs are approximately equal) see e.g. [81], where strong hardness results on convergence times of best response dynamics are drawn, [21], where positive results for the convergence times in symmetric games are given, [9], where the above work is extended to the asymmetric case when a special property that helps to bypass the inapproximability of [81] holds and [1] where properties of strategies' spaces are investigated so as to have fast convergence times.

For games with player-specific cost functions see e.g. [63], where the study of player-specific cost functions was initiated in singleton CGs, [2], where, among others, more general networks that guarantee PNE existence are investigated, [61], where the differences in the player-specific cost functions that players perceive are only constants, [2], where the networks found in [61] are proved to be optimal topologically and a polynomial algorithm of finding the guaranteed to exist pure PNE is given and [46] where properties of the class of cost functions are investigated for both the cases of weighted or unweighted CGs in order to have PNE.

For games with weighted players see e.g. [44], where the model of weighted CGs is introduced and results on the existence of equilibrium and potential functions are presented, [69] for experimental results on computing PNE in the model of [44], [2] where, as in player-specific games, more general networks that guarantee PNE existence are investigated, [14], where the PoA (whenever a PNE exists) is related with the class of allowable cost functions and worst case results are proved in close relation with the ones in unweighted games, [65], where topological properties are shown to strongly relate with the existence of a PNE and [50], where an exact characterization of the set of cost functions that guarantee a PNE existence is given.

Bottleneck Costs

For results in non atomic games see e.g. [27], where the theoretical study of bottleneck games were initiated and [62], where the uniqueness and optimality of PNE is investigated in parallel-links, series parallel and general networks. For results concerning Braess's paradox see section 1.3.

For results in atomic games see e.g. [11], where the study for atomic games was initiated, the non uniqueness of PNE was proved, convergence's rates were studied and bounds on PoA were achieved, [18], where results on the PoA related to the network structure are presented while existence of a social optimal PNE is also proved, [31], where complete characterizations of the networks that have optimal PoA is given (series-parallel networks for the pre-described version of bottleneck games and extended parallel net-

works for another version defined therein) and [19], where results related to the existence, complexity, and price of anarchy of PNE for several network games (symmetric and asymmetric, with identical or weighted players) are given.

1.3 Reducing the Price of Anarchy

The degradation of the network due to players' selfish behavior has drawn much attention in the past. In the literature, there are three prevailing ways to tackle this degradation.

Taxing the edges: changing the cost functions of the players by adding taxes on the edges (resources) of the network.

Stackelberg strategies: changing the fraction of selfish players by assuming that some players are willing to cooperate for social welfare.

Excluding the Braess's paradox: changing the network topology by making some edges (resources) unavailable.

1.3.1 Taxing the Edges of the Network

A CG with taxes is a typical CG with an extra tax vector that gives the taxes for the edges of the network. The players' cost are modified so as to consider also the cost due to taxes on the edges they choose. Players may have different sensitivities in taxes and this is captured by a constant coming along with each player.

Taxes increase the cost of the players without affecting the social cost function and thus can be used without charge by the network manager. Ideally, one would like, by taxing the edges, to have the Nash Equilibria of the new game be an optimum flow in the original game. These taxes are called *optimal taxes*.

Non Atomic Case

In non atomic games the problem of designing optimal tax vectors has been studied extensively. A classic result going all the way back to Pigou [70] states that marginal cost taxes induce the optimal traffic pattern for homogeneous players [13]. A significant volume of recent work on optimal taxes for non atomic CGs considers the more intriguing and realistic case of *heterogeneous* players, which may have different valuations of time (latency)

in terms of money (taxes). Yang and Huang [86] established the existence of optimal taxes for non atomic asymmetric network CGs with heterogeneous players. Subsequently, their result was rediscovered by Fleischer, Jain, and Mahdian [36], and Karakostas and Kolliopoulos [54]. Previously the single-source special case had been investigated by Cole, Dodis, and Roughgarden [26].

The existence of optimal taxes for non atomic CGs with heterogeneous players follows from Linear Programming duality ([36, 54]), and thus an optimal tax vector can be computed efficiently by solving a linear program.

For non atomic games, under mild assumptions on the latency functions the edge flow at equilibrium is unique. Hence the taxes of [13, 26, 36, 54, 86] induce the optimal solution as the unique edge flow of the equilibria of the game with taxes.

Atomic Case

Atomic CGs, even with splittable traffic (players can use more than one paths to route their flow), may admit many different Nash equilibria, possibly with different edge flows. Therefore, when considering atomic games, one has to distinguish between

- *weakly-optimal* tax vectors, for which at least one Nash equilibrium of the game with taxes minimizes the total latency, and
- *strongly-optimal* tax vectors, for which all Nash equilibria of the game with taxes minimize the total latency.

For atomic CGs with splittable traffic and heterogeneous players, Swamy [82] proved that weakly-optimal tax vectors exist and can be computed efficiently. In fact they can be computed by solving a convex program similar to the ones in [36] and [54].

As for atomic CGs with unsplittable traffic, the existence and efficient computation of optimal taxes has been studied in the setting of homogeneous players.

Caragiannis, Kaklamanis, and Kanellopoulos [20] considered atomic games with linear latency functions and homogeneous players, and investigated how much taxes can improve the price of anarchy. On the negative side, they established that if the players either do not share the same source and sink or have different traffic demands, then strongly-optimal taxes may not exist. On the positive side, they presented an efficient construction of strongly-optimal taxes for parallel-link games with linear latencies and unit-demand players.

Subsequently, Fotakis and Spirakis [45] proved that weakly-optimal taxes exist and can be computed efficiently for atomic symmetric network CGs, and that such taxes are strongly-optimal if the network is series-parallel.

For the case of heterogeneous players, results follow from the technique used in the non atomic case. Fotakis, Karakostas and Kolliopoulos in [43], using a linear program like one in [54], prove the existence of weakly optimal tolls for games with heterogeneous players with the *same source node*. Also, in [43], it is given a counter example that shows that players' heterogeneity precludes the existence of strongly-optimal taxes even on the simplest topology of parallel-link networks.

Restrictions on the Allowable Taxes

More recently, Höfer et al. ([51]) studied non atomic CGs with taxes where only a subset of the resources is allowed to get taxes and, on the negative side they provided an NP-hardness result for finding optimal taxes for general networks with linear latency functions and two commodities while on the positive side, for single-commodity networks with parallel links and linear latency function, they provided a polynomial time algorithm for finding optimal taxes.

Following this work, Bonifaci et al. ([17]) studied the (non atomic) case where along with each edge, an upper bound on the allowable tax on that edge is given. They provide a characterization for the flows that can be imposed by the restricted taxes and compute the optimal taxes when the optimal flow is inducible. Also based on this characterization, they manage to compute the taxes that induce the smallest cost at equilibrium for parallel links networks. They also derive tight (even for parallel link networks) bounds on the efficiency of restricted tolls for multi-commodity networks and polynomial latency functions.

Jelinek et al. in [53] generalized the above model to atomic CGs and to CGs with taxes with heterogeneous players. For non atomic and heterogeneous players, they prove that the problem is NP-hard even for single-commodity networks and affine latency functions. On the positive side they give an algorithm for optimally taxing subnetworks with affine latency functions. For weighted atomic players, the problem is NP-hard already for parallel-arc networks and linear latency functions, even if players are homogeneous. Focusing on parallel links games, for unweighted atomic and homogeneous players, they develop an algorithm to compute optimal restricted tolls and for unweighted atomic and heterogeneous players, they derive an algorithm for optimally taxing subnetworks.

1.3.2 Stackelberg Strategies

In Stackelberg routing, the network manager, as a central authority, coordinates a fixed fraction a of the players and assigns them to appropriately selected strategies trying to minimize the performance degradation due to the selfish behavior of the remaining players.

A *Stackelberg strategy* is an algorithm that determines the strategies of the coordinated players. The problem that arises for the coordinating authority is, given a fraction a under her influence, to find the *best stackelberg strategy* so as to minimize the inefficiency caused by the selfishness of the other players. Note that for $a = 0$ we get a classical congestion game.

The first case that has been thoroughly studied is the case of parallel link networks (networks with two nodes and parallel edges joining them) with linear latency functions.

Roughgarden in [75] proved that it is NP-hard to compute the optimal Stackelberg strategy via a technical reduction from the $\frac{1}{3}\frac{2}{3}$*PARTITION* problem. He also presented three simple strategies with provable performance guarantees. The first of the three, computes the optimum of the game with *an* players (n the number players in the original game) and finds the edges for these players. In the original game the algorithm assigns the coordinated players on the same edges they would stay in the optimum he computed. The second one, namely the *SCALE* algorithm, after computing the optimal solution x^* assign to each link e, ax_e^* players. The third one, the Largest Latency First (*LLF*) algorithm, computes the optimal solution and assigns the coordinated players to the most costly edges (links), fulling them in decreasing order.

Swamy in [82] obtained the first results for non atomic routing in graphs more general than parallel-link graphs, and strengthen existing results for parallel-link graphs. In series-parallel graphs (sepa) he showed that Stackelberg routing reduces the PoA to a constant (depending on the fraction of flow controlled). The algorithm that does the work is the *LLF*. For general graphs, he obtained latency-class specific bounds on the PoA with Stackelberg routing, which give a continuous trade-off between the fraction of flow controlled and the PoA. The bounds come from analyzing the performance of the *SCALE* and the *LLF* algorithm. A part of this proof followed the same technique as the one in Correa et. al ([28]). In parallel-link graphs, he showed that Stackelberg routing reduces the PoA to a convex combination of the worst case PoA when $a = 0$ and the optimal performance when $a = 1$. The convex combination factors are $1 - a$ and a repsectively.

Karakostas and Kolliopoulos in [54] analyzed *LLF* and *SCALE* algorithms, for general topology networks, multicommodity players, and linear

latency functions. They showed a PoA bound for *SCALE* which decreases from worst case PoA to 1 as a increases from 0 to 1, and depends only on a, generalizing this way the known bound for the parallel links network which nevertheless has one single commodity. An interesting fact is that a good lower bound for *SCALE* is the instance of Braess's Paradox which we'll see later on (section 1.3.3). A weaker bound for *LLF* and some extensions to general latency functions were also shown.

Bonifaci et al. in [16], constructed a family of singlecommodity (non atomic) instances such that every Stackelberg strategy induces a price of anarchy that grows linearly with the size of the network. This bound does not depend on the fraction a of the coordinated players. Moreover, they prove upper bounds on the price of anarchy of the largest-latency-first (LLF) strategy that only depend on the size of the network. Besides other implications, this rules out the possibility to construct constant-size networks to prove an unbounded price of anarchy. They also analyze the effectiveness of SCALE, proving bounds for a general class of latency functions that includes polynomial latency functions as a special case. Their analysis is based on an approach that is simple yet powerful enough to obtain (almost) tight bounds for SCALE in general networks improving this way the bounds of Swamy ([82]) presented above

Fotakis in [38] investigated the effectiveness of Stackelberg strategies for atomic CGs with unsplittable demands. Two orthogonal cases were considered: i) linear CGs with arbitrary strategies and ii) CGs on parallel links with arbitrary non-negative and non-decreasing latency functions. For the second case, the same bound as in non atomic games (Swamy [82]) was proved. For the case of linear CGs with arbitrary strategies, algorithms *SCALE* (changed appropriately for the atomic case) and *LLF* were analyzed and upper and lower bounds were derived

In [38], a new stackelberg strategy was proposed. An interesting case arises when the number of players is large and the number of coordinated players is considerably larger than the number of resources, even if a is small. To take advantage of this possibility, a simple Stackelberg strategy was introduced, called *COVER*. Assuming that the ratio of the number of coordinated players to the number of resources is no less than a positive integer λ, *COVER* assigns to every resource either at least λ or as many coordinated players as the resource has in the optimal configuration. The PoA of *COVER* tends to the PoA of the corresponding non atomic linear CG as λ grows. The idea proposed was to combine algorithms *LLF* and *SCALE* with *COVER* to take advantage of the above fact. On the negative side, in [38], it is presented a lower bound that holds not only for *SCALE*, but

also for any randomized Stackelberg strategy that assigns the coordinated players to their optimal strategies.

1.3.3 Tackling the Braess's Paradox

Braess's Paradox is a counter intuitive fact stating that when removing (adding) edges from (to) a network, its performance may increase (decrease). Figures 1.2 and 1.3 are examples for additive and bottleneck costs games respectively.

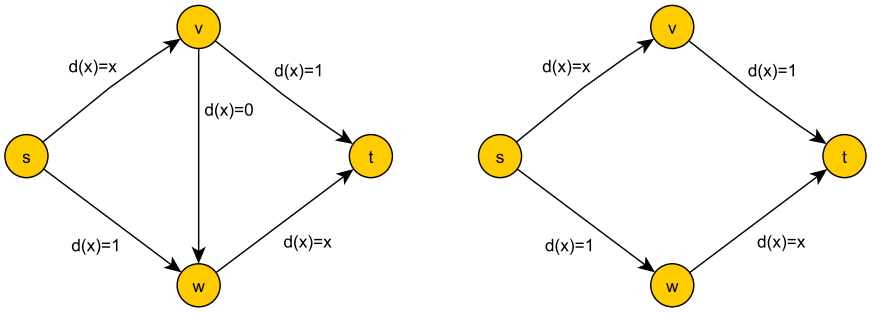

Figure 1.2: A unit of flow is to be routed from s to t. (a). The optimal (additive costs) flow routes $1/2$ unit of traffic on the upper path (s, v, t) and $1/2$ unit on the lower path (s, w, t), and achieves a total latency of $3/2$. In the Nash flow, all traffic goes through the path (s, v, w, t). The players' latency is 2, and the PoA is $4/3$. (b). Without the edge (v, w), the Nash flow coincides with the optimal flow.

The idea is to improve the network performance at equilibrium by exploiting the essence of the Braess's paradox, that is to remove some network edges in order to decrease the latency of the Nash flow (the induced flow on Nash equilibrium). Thus, given a CG, we seek for the *best subnetwork*, i.e. the subnetwork minimizing the players' latency at (worst) equilibrium.

Valiant and Roughgarden [84], in order to theoretically support that Braess's paradox is not an artifact of theory, proved that (under additive costs) it occurs with high probability on random networks, and that for a natural distribution of linear latencies, edge removal may improve, with high probability, the equilibrium latency by a constant factor. Across the same lines, Chung and Young in [24] adopting the random graphs model of Erdös-Rényi showed that Braess's paradox occurs when $np \geq c \log(n)$ for some $c > 1$ (n, p the random graphs' parameters). See also [47] for a slight generalization by the same authors.

Additive Costs Games

Unfortunately, Roughgarden [78] proved that it is *NP*-hard not only to find the best subnetwork, but also to compute any meaningful approximation to the equilibrium latency on the best subnetwork. In particular, he showed that even for linear latencies, it is *NP*-hard to distinguish between *paradox-free* instances, where edge removal cannot improve the equilibrium latency, and *paradox-ridden* instances, where the total latency of the Nash flow on the best subnetwork is equal to the optimal total latency (i.e. edge removal can decrease the PoA to 1). This implies that the only known algorithm for approximating the equilibrium latency on the best subnetwork is the trivial one, which does not remove any edges!

Fotakis, Kaporis and Spirakis in ([42]) examined the "Braess's paradox problems" for some practically interesting settings and managed to provide a polynomial-time algorithm that decides if a network is paradox-ridden, when latencies are linear and strictly increasing, a polynomial-time algorithm for the problem of finding the best subnetwork, which outperforms any known approximation algorithm for the case of strictly increasing linear latencies and an algorithm for finding a subnetwork that is almost optimal wrt equilibrium latency which is *subexponential* when the number of paths is polynomial and each path is of polylogarithmic length.

They also prove that the problem of deciding if a network with arbitrary linear latencies is paradox-ridden reduces to the problem of generating all optimal basic feasible solutions of a Linear Program that describes the optimal traffic allocations to the edges with constant latency. As an extension of exploiting the paradox, it was provided a polynomial-time method that turns the optimal flow into a Nash flow by deleting the edges not used by the optimal flow, and performing minimal modifications to the latencies of the remaining ones.

Since Roughgarden's negative results, trying to detect Braess's Paradox wasn't incentive. Considering the results of [42] though, the problem of detecting the "bad", Braess's paradox's edges, if any, gets a lot more interesting.

Bottleneck Costs Games

Every bottleneck routing game is known to admit a Nash flow that is optimal for the network, in the sense that it minimizes the maximum latency on any used edge, a.k.a. the bottleneck cost of the network (see e.g. [11, Corollary 2]). On the other hand, bottleneck routing games usually admit many different Nash flows, some with a bottleneck cost quite far from the

optimum. Hence, there has been a considerable interest in quantifying the performance degradation due to the players' non-cooperative and selfish behavior in (several variants of) bottleneck routing games.

Simple examples (see, e.g., figure 8.2 or [27, Figure 2]) demonstrate that the PoA of bottleneck routing games with linear latency functions can be as large as $\Omega(n)$, where n is the number of vertices of the network.

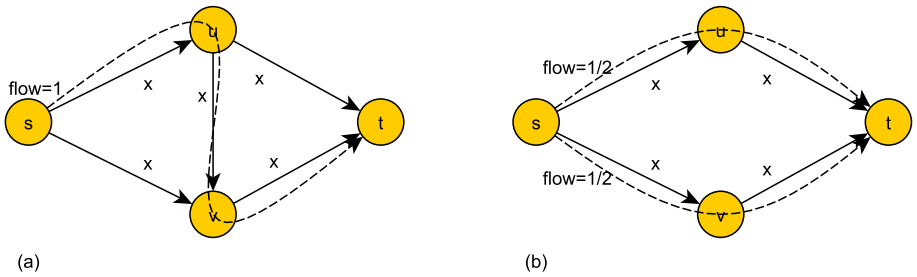

Figure 1.3: An example of Braess's paradox for bottleneck routing games. We consider a routing instance with identity latency functions and a unit of traffic to be routed from s to t. The worst Nash flow, in (a), routes all flow through the path (s, u, v, t), and has a bottleneck cost of 1. On the other hand, the optimal flow routes $1/2$ unit through the path (s, u, t) and $1/2$ unit through the path (s, v, t), and has a bottleneck cost of $1/2$. Hence, PoA = 2. In the subnetwork (b), obtained by removing the edge (u, v), we have a unique Nash flow that coincides with the optimal flow, and thus the PoA becomes 1.

For atomic splittable bottleneck routing games, where the population of players is finite, and each player controls a non-negligible amount of traffic which can be split among different paths, Banner and Orda [11] observed that the PoA can be unbounded, even for very simple networks, if the players have different origins and destinations and the latency functions are exponential. On the other hand, Banner and Orda proved that if the players use paths that, as a secondary objective, minimize the number of bottleneck edges, then all Nash flows are optimal. For a variant of non atomic bottleneck routing games, where the social cost is the average (instead of the maximum) bottleneck cost of the players, Cole, Dodis, and Roughgarden [27] proved that the PoA is $4/3$, if the latency functions are affine and a subclass of Nash flows, called *subpath-optimal Nash flows*, is only considered. Subsequently, Mazalov et al. [62] studied the inefficiency of the best Nash flow under this notion of social cost.

For atomic unsplittable bottleneck routing games, where each player routes a unit of traffic through a single $s - t$ path, Banner and Orda [11] proved that for polynomial latency functions of degree d, the PoA is $O(m^d)$, where m is the number of edges of the network. On the other hand, Epstein

et al. ([31]) proved that for series-parallel networks with arbitrary latency functions, all Nash flows are optimal. Subsequently, Busch and Magdon-Ismail [18] proved that the PoA of atomic unsplittable bottleneck routing games with identity latency functions can be bounded in terms of natural topological properties of the network. In particular, they proved that the PoA of such games is bounded from above by $O(l + \log n)$, where l is the length of the longest $s - t$ path, and by $O(k^2 + \log^2 n)$, where k is length of the longest cycle.

Once again, in this setting, one may distinguish two extreme classes of instances: *paradox-free* instances, where edge removal cannot improve the bottleneck cost of the worst Nash flow, and *paradox-ridden* instances, where the bottleneck cost of the worst Nash flow in the best subnetwork is equal to the optimal bottleneck cost of the original network (see also [78, 42]).

The approximability of selective network design, a generalization of network design where we cannot remove certain edges, was considered by Hou and Zhang [52]. For atomic unsplittable bottleneck routing games with a different traffic rate and a different origin and destination for each player, they proved that if the latency functions are polynomials of degree d, it is NP-hard to approximate selective network design within a factor of $O(m^{d-\varepsilon})$, for any constant $\varepsilon > 0$. Moreover, for atomic k-splittable bottleneck routing games with multiple origin-destination pairs, they proved that selective network design is NP-hard to approximate within any constant factor.

However, a careful look at the reduction of [52] reveals that their strong inapproximability results crucially depend on both (i) that we can only remove certain edges from the network, so that the subnetwork actually causing a high PoA cannot be destroyed, and (ii) that the players have different origins and destinations (and also are atomic and have different traffic rates). As for the importance of (ii), in a different setting, where the players' individual cost is the sum of edge latencies on their path and the social cost is the bottleneck cost of the network, it is known that Braess's paradox can be dramatically more severe for instances with multiple origin-destination pairs than for instances with a single origin-destination pair. More precisely, Lin et al. [59] proved that if the players have a common origin and destination, the removal of at most k edges from the network cannot improve the equilibrium bottleneck cost by a factor greater than $k + 1$. On the other hand, Lin et al. [58] presented an instance with two origin-destination pairs where the removal of a single edge improves the equilibrium bottleneck cost by a factor of $2^{\Omega(n)}$. Therefore, both at the technical and at the conceptual level, the inapproximability results of [52] do not really shed light on the approximability of the (simple, non-selective)

network design problem in the simplest, and most interesting, setting of non atomic bottleneck routing games with a common origin-destination pair for all players.

1.4 Stochastic Congestion Games

Most research work on CGs essentially ignores the stochastic nature of edge delays, a feature of most practical applications, and assumes that the players select their strategies based on precise knowledge of the edge latencies, which are considered to be deterministic. On the contrary, in real life transportation or telecommunication networks, the players cannot accurately predict the actual edge delays. This happens not only because the players cannot know the exact congestion of every edge, but also due to (a priori unknown) external events (e.g., some construction work, a minor accident, a link failure) that may affect the edge latencies and introduce uncertainty. It is therefore natural to assume that the players decide on their strategies based only on estimations of their actual delay, and most important, that they are fully aware of the uncertainty and of the potential inaccuracy of their estimations. So, to secure themselves from the event of an increased delay, whenever this may have a considerable influence, the players select their paths taking uncertainty into account (e.g., people either take a safe route or plan for a larger-than-usual delay when they head to an important meeting).

Such considerations give rise to CGs with stochastic delays and risk-averse players, where instead of the path that minimizes her predicted (or expected) delay, each player selects a path that guarantees her a reasonably low actual delay with a reasonably high confidence. To take uncertainty into account, the actual delay of each player can be modeled by a random variable. Then, a common assumption is that the players seek to minimize either a convex combination of the expectation and the variance of their delay, or a player-specific quantile of the delay distribution (see also [83, 34] about the cost functions of risk-averse players, and [72] about possible ways of risk quantification in optimization under uncertainty).

Following the research direction above, Ordóñez and Stier-Moses [68] considered non atomic CGs, where the population of players is infinite and each player controls a negligible amount of load, and suggested that each path should be penalized by an additive term that increases with the risk-aversion of the players and with the maximum deviation from the expected delay of the path (however, this term does not depend on the actual load of the edges). For each path, the additive term can be chosen either as a

δ-fraction of (resp. a δ-quantile of a random variable depending on) the maximum deviation from the expected delay of the path, or simply, as the sum of the δ-fractions of the maximum deviation from the expected delay of each edge in the path, where δ quantifies the risk-aversion of the players. Under some general assumptions, [68] proves that an equilibrium exists and is essentially unique in all the cases above.

Subsequently, Nikolova and Stier-Moses [67] suggested a model of stochastic selfish routing with risk-averse players, where each player selects a path that minimizes the expected delay plus δ times the standard deviation of the delay, where δ quantifies the risk-aversion of the players. They considered non atomic and atomic CGs, mostly with homogeneous players, that share the same risk attitude, and distinguished between the case where the standard deviation of a path's delay is *exogenous*, i.e., it does not depend on the load of the edges in the path, and the case where the standard deviation is *endogenous*, i.e., it is a function of the load. Nikolova and Stier-Moses [67] proved that in the exogenous case, which is similar to the model of [68], (non atomic and atomic) stochastic routing games essentially retain the nice properties of standard CGs: they admit a potential function and, in the non atomic setting, a unique equilibrium, and the inefficiency of equilibria can be bounded as for standard CGs. In the endogenous case, they proved that non atomic stochastic routing games admit an equilibrium, which is not necessarily unique, but may not admit a cardinal potential. Moreover, atomic stochastic routing games may not admit a PNE even in simple extension-parallel networks with 2 players and linear delays.

Following this research agenda, a better understanding of the properties of (atomic) CGs with stochastic delays and risk-averse players seems meaningful while cases with different risk-aversions for the players have not been studied at all.

Actually, there is a significant volume of work on theoretical and practical aspects of transportation networks with uncertain delays (see e.g., the discussion and the references in [67]). However, that line of research focuses on the non atomic setting and adopts specific notions of individual cost and viewpoints. Motivated from applications where the players have only partial knowledge of the number of players participating in the game, Ashlagi, Monderer, and Tennenholtz [7] considered CGs on parallel links with stochastic players. However, the players' individual cost in their model is the expected delay of the link chosen, and thus the players are risk-neutral. They studied mixed Nash equilibria, and proved that a generalization of the fully mixed equilibrium remains a mixed equilibrium if the players are stochastic.

Chapter 2

Contribution

Congestion Games constitute one of the most extensively studied fields of Algorithmic Game Theory. Nevertheless, as expected, there are still open problems in this area.

Our focus was twofold, yet finally connected. In one direction we dealt with problems related to detecting and excluding the Braess's Paradox, both in additive costs and bottleneck costs games, which lie in the family of problems related to reducing the Price of Anarchy of CGs.

In the other direction we addressed with the problem of generalizing CGs in such a way that uncertainty on the delays is taken under consideration, focusing basically on modeling the source of uncertainty on the resources. Apart from the drawn results for this model, surprisingly, this modeling shed light to a new way for improving the Price of Anarchy of a network, i.e. abuse the players' risk averse behavior and improve the network's performance by adding extra uncertainty as "noise" in its resources. By considering this, the directions of our work get an immediate relation.

Next we outline our contribution in each of the problems we investigated, while in the following sections we get into more detail for each of them.

- Braess Paradox in (the different versions of) bottleneck routing games was not fully understood and we managed to clear the field for the basic version of them (subsection 2.1 and chapter 4).

- Random networks (considered thus far in the literature) with high probability suffer from the Braess's Paradox when costs are additive. We managed to prove essentially that the way to prove the above can be cleverly yet carefully used to efficiently exploit the paradox in those networks (subsection 2.2 and chapter 5).

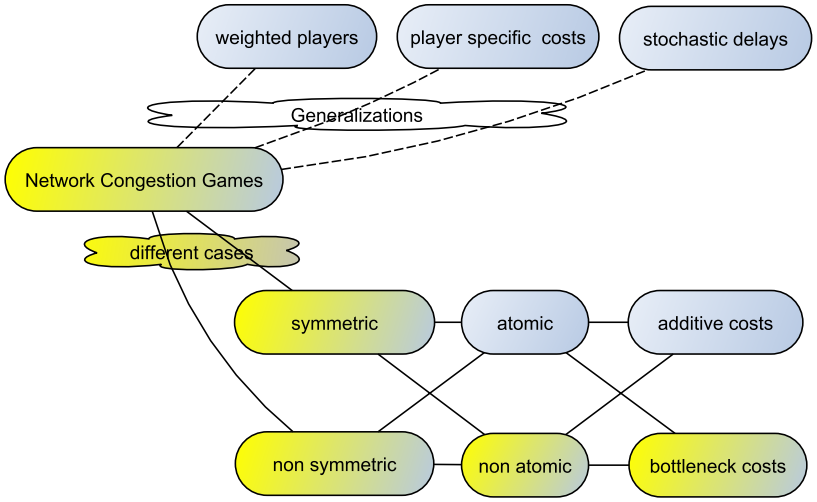

Figure 2.1: The main results (hardness results) of our work apply to both symmetric and asymmetric non atomic bottleneck costs games.

- Stochastic CGs and risk aversion has drawn much attention in the very recent years. By the work in chapter 6 (contribution in subsection 2.3), we move one step forward on understanding these classes of games.
- In certain settings, stochastic delays can actually improve the network performance at equilibrium (see e.g. [71]). In chapter 7 (contribution in 2.4), we show how we can abuse the risk aversion of the players and improve the network's performance.

2.1 Braess's Paradox in Bottleneck Costs Games

We investigate the approximability of the network design problem for the simplest, and seemingly easier to approximate, variant of non-atomic bottleneck routing games (with a single origin-destination pair). Our main result is that network design is hard to approximate within reasonable factors, and holds even for the special case of strictly increasing linear latencies. To the best of our knowledge, this is the first work that investigates the impact of Braess's paradox and the approximability of the network design problem for the basic variant of bottleneck routing games. For the original work see [41]. Figure 2.1 captures the model on which the results of this work hold.

In Section 4.2, we use techniques similar to those in [31, 27], and show that bottleneck routing games do not suffer from Braess's paradox either if the network is series-parallel, or if we consider only subpath-optimal Nash flows (definition in section 4.1).

On the negative side, we employ, in Section 4.3, a reduction from the 2-Directed Disjoint Paths problem, and show that for linear bottleneck routing games, it is NP-hard to recognize paradox-ridden instances (Lemma 4.1). In fact, the reduction shows that it is NP-hard to distinguish between paradox-ridden instances and paradox-free instances, even if their PoA is equal to 4/3, and thus, it is NP-hard to approximate the network design problem within a factor less than 4/3.

In Section 4.4, we apply essentially the same reduction, but in a recursive way, and obtain a much stronger inapproximability result. In particular, we assume the existence of a γ-gap instance, which establishes that network design is inapproximable within a factor less than γ, and show that the construction of Lemma 4.1, but with some edges replaced by copies of the gap instance, amplifies the inapproximability threshold by a factor of 4/3, while it increases the size of the network by roughly a factor of 8 (Lemma 4.4). Therefore, starting from the 4/3-gap instance of Lemma 4.1, and recursively applying this construction a logarithmic number times, we show that it is NP-hard to approximate the network design problem for linear bottleneck routing games within a factor of $O(n^{0.121-\varepsilon})$, for any constant $\varepsilon > 0$. An interesting technical point is that we manage to show this inapproximability result, even though we do not know how to efficiently compute the worst equilibrium bottleneck cost of a given subnetwork. Hence, our reduction uses a certain subnetwork structure to identify good approximations to the best subnetwork. To the best of our knowledge, this is the first rime that a similar recursive construction is used to amplify the inapproximability threshold of the network design problem, and of any other optimization problem related to selfish routing.

In Section 4.5, we consider latency functions that satisfy a Lipschitz condition, and present an algorithm for finding a subnetwork that is almost optimal w.r.t. the bottleneck cost of its worst Nash flow, when the worst Nash flow in the best subnetwork routes a non-negligible amount of flow on all used edges. The algorithm is based on Althöfer's Sparcification Lemma [5], and is motivated by its recent application to network design for additive routing games [42]. For any constant $\varepsilon > 0$, the algorithm computes a subnetwork and an $\varepsilon/2$-Nash flow whose bottleneck cost is within an additive term of $O(\varepsilon)$ from the worst equilibrium bottleneck cost in the best subnetwork. The running time is roughly $|\mathcal{P}|^{\text{poly}(\log m)/\varepsilon^2}$, and is quasipolynomial, when the number $|\mathcal{P}|$ of paths is quasipolynomial.

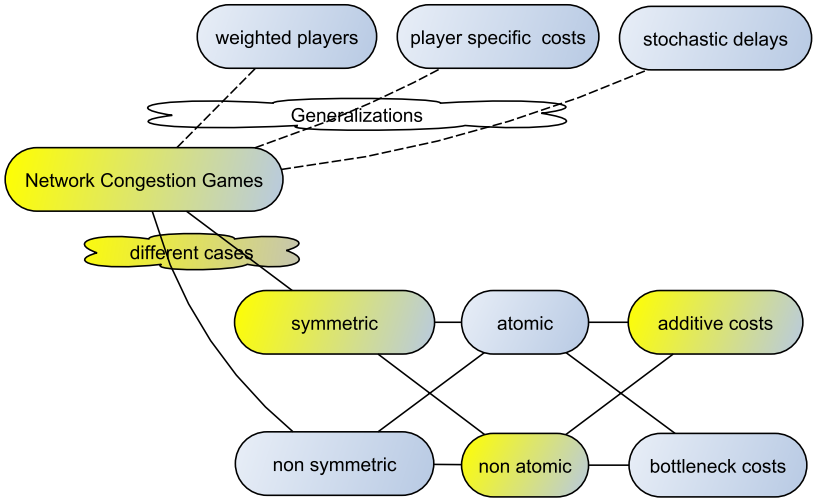

Figure 2.2: The case where the random networks under study lay

2.2 Braess's Paradox in Additive Costs Games

Departing from [24, 25, 84], that proved that random Erdös-Rényi $\mathcal{G}_{n,p}$ graphs are prone to the paradox, we adopt a purely algorithmic approach. We focus on the class of so-called *good* selfish routing instances, namely instances with the properties used by [24, 84] to demonstrate the occurrence of Braess's paradox in random networks with high probability. In fact, one can easily verify that the random instances of [24, 84] are good with high probability. Rather surprisingly, we prove that, in many interesting cases, we can efficiently approximate the best subnetwork and its equilibrium latency.

What may be even more surprising is that our approximation algorithm is based on the expansion property of good instances, namely the very same property used by [24, 84] to establish the prevalence of the paradox in good instances! To the best of our knowledge, our results are the first of theoretical nature which indicate that Braess's paradox can be efficiently eliminated in a large class of practically interesting instances. For the original work see [40]. Figure 2.2 captures the model considered in this work.

Technically, we present essentially an approximation scheme; given a good instance and any constant $\varepsilon > 0$, we compute a flow g that is an ε-Nash flow for the subnetwork consisting of the edges used by it, and has a latency of $L(g) \leq (1+\varepsilon)L^* + \varepsilon$, where L^* is the equilibrium latency of the best subnetwork (Theorem 5.1). Flow g whp has these properties.

Our results hold for any network in the class of good networks. This, of course, includes $\mathcal{G}_{n,p}$ with p above the connectivity threshold, but it might also include other types of random expanders. Our approximation scheme runs in polynomial time for the most interesting case that the network is relatively sparse and the traffic rate r is $O(\text{poly}(\ln \ln n))$, where n is the number of vertices. Specifically, the running time is polynomial if the good network has average degree $O(\text{poly}(\ln n))$, i.e., if $pn = O(\text{poly}(\ln n))$, for random $\mathcal{G}_{n,p}$ networks, and quasipolynomial for average degrees up to $o(n)$. As for the traffic rate, most work on selfish routing and selfish network design problems assumes that $r = 1$, or at least that r does not increase with the network's size (see e.g., [77], and the references there in). So, we can approximate, in polynomial-time, the best subnetwork for a large class of instances that, with high probability, include exponentially many $s - t$ paths and paths of length $\Theta(n)$. For such instances, a direct application of [42, Theorem 3] gives an exponential-time algorithm.

The main idea behind our approximation scheme, and our main technical contribution, is a polynomial-time approximation-preserving reduction of the best subnetwork problem for a good network G to a corresponding best subnetwork problem for a 0-*latency simplified network* G_0, which is a layered network obtained from G if we keep only s, t and their immediate neighbors, and connect all neighbors of s and t by direct edges of 0 latency. We first show that the equilibrium latency of the best subnetwork does not increase when we consider the 0-latency simplified network G_0 (Lemma 5.2). Although this may sound reasonable, we highlight that decreasing edge latencies to 0 may trigger Braess's paradox (e.g., starting from the network in Fig. 5.1.a with $d_{(v,w)}(x) = 1$, and decreasing it to $d_{(v,w)}(x) = 0$ is just another way of triggering the paradox). The importance of our 0-*latency simplified network* is that it greatly simplifies the network design problem, since it allows us to focus on the loads of the s, t incident edges. In sharp contrast, the corresponding subnetworks in [84, Fig. 3b], [25, Sect. 2.3 Fig. 2], implicitly exhibit the paradox: they apply Chernoff's bounds to show that appropriate parts of these subnetworks whp have large cardinalities, implying that even more flow r can be routed through these subnetworks without increasing the selfish latency of the original network. The paradox implicitly follows by an non obvious yet intuitive fact [49, 59]: the selfish common latency is strictly increasing with the total flow r. Hence, we approximate the best subnetwork problem (Theorem 5.7) for our 0-latency simplified network by employing an approximate version of Caratheodory's Theorem (Theorem 5.6).

The final (and crucial) step of our approximation preserving reduction is to start with the flow-solution to the best subnetwork problem for the

0-latency simplified network, and extend it to a flow-solution to the best subnetwork problem for the original (good) instance. To this end, we show how to "simulate" 0-latency edges by low latency paths in the original good network. Intuitively, this works because due to the expansion properties and the random latencies of the good network G, the intermediate subnetwork of G, connecting the neighbors of s to the neighbors of t, essentially behaves as a complete bipartite network with 0-latency edges. This is also the key step in the approach of [24, 84], showing that Braess's paradox occurs in good networks with high probability (see [24, Section 2]). Hence, one could say that to some extent, the reason that Braess's paradox exists in good networks is the very same reason that the paradox can be efficiently resolved. Though conceptually simple, the full construction is technically involved and requires dealing with the amount of flow through the edges incident to s and t and their latencies. Our construction employs a careful grouping-and-matching argument, which works for good networks with high probability, see Lemmas 5.8 and 5.9.

We highlight that the reduction itself runs in polynomial time. The time consuming step is the the one returning the (approximate) solution to the 0-latency simplified network. Since such networks have only polynomially many (and very short) $s-t$ paths, they escape the hardness result of [78]. The approximability of the best subnetwork for 0-latency simplified networks is an intriguing open problem arising from our work that we discuss in section 8.1.

Our result shows that a problem, that is NP-hard to approximate, can be very closely approximated in random (and random-like) networks. This resembles e.g., the problem of finding a Hamiltonian path in Erdös-Rényi graphs, where again, existence and construction both work just above the connectivity threshold, see e.g., [15]. However, not all hard problems are easy when one assumes random inputs (e.g., consider factoring or the hidden clique problem, for both of which no such results are known in full depth).

2.3 Stochastic Congestion Games

We restrict our attention to atomic CGs, and introduce two variants of stochastic CGs, which are inspired by the main sources of uncertainty in delays of real transportation and telecommunication networks. We start from the observation that the variability of edge delays comes either from the variability of the traffic demand, and the subsequent variability of the edge loads, or from the variability of the edge performance level. Decou-

pling them, we introduce two variants of stochastic CGs, namely *Congestion Games with Stochastic Players* and *Congestion Games with Stochastic Edges*, each capturing one of the two sources of uncertainty above. For the original work see [6]. Figure 2.3 places our models on the congestion games map.

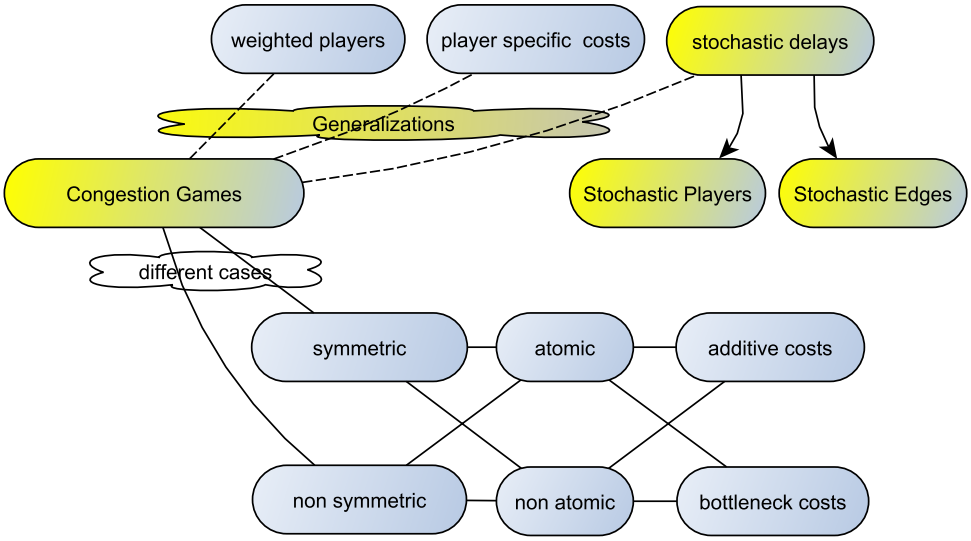

Figure 2.3: The introduced models and their positioning on the map.

CGs with Stochastic Players aim to model the variability of the traffic demand. Specifically, each player i participates in the game, by actually traversing her path, independently with probability p_i. As a result, the total network load, the edge loads, and the edge and the path latencies are all random variables. On the other hand, CGs with Stochastic Edges aim to model variability in the network operation. Now, each edge e may operate either at the "standard" mode, where its latency is given by a function $f_e(x)$, or at the "faulty" mode, where its latency is given by $g_e(x)$, where $g_e(x) \geq f_e(x)$ for all $x \geq 0$ (e.g., an edge operates at the "faulty" mode after a minor accident or a link failure). Each edge e switches to the "faulty" mode independently with a given probability p_e. Hence, the network load and the edge loads are now deterministic, but the edge and the path latencies are random variables. In both variants, we assume that the players adopt a risk-averse attitude to the stochastic delays. Specifically, each player i has a (possibly different) desired *confidence level* δ_i, and her individual cost on a path q is the δ_i-quantile (a.k.a. value-at-risk) of the delay distribution of q. In words, the individual cost of player i is the minimum delay she can achieve along q with probability at least δ_i.

At the conceptual level, the model of CGs with Stochastic Players is similar to the model with endogenous standard deviations of [67]. In fact, using Chernoff bounds, we can show that for linear latency functions, if the expected edge loads are not too small, our δ_i-quantile individual cost can be approximated by the individual cost used in [67]. However, we also consider stochastic demands, a direction suggested in [67, Sec. 7] to enrich the model, and players that are heterogeneous with respect to risk attitude. As for CGs with Stochastic Edges, the model is conceptually similar to the model with exogenous standard deviations of [67].

In the technical part, we restrict ourselves to the important special case of parallel-link networks with symmetric player strategies, and investigate how the properties of stochastic CGs depend on whether the players have the same participation probabilities and/or confidence levels or not. We first observe that such games admit a potential function and an efficiently computable PNE, if the players are homogeneous, namely if they have the same confidence level δ and, in case of stochastic players, the same participation probability p (Theorems 6.1 and 6.8). We also show that if the players have different confidence levels (and the same participation probability, if they are stochastic), stochastic CGs belong to the class of player-specific congestion games [63], and thus admit a PNE computable in polynomial time (Corollaries 6.2 and 6.9). On the negative side, we prove that such games do not admit a potential function (Theorems 6.3 and 6.10). For CGs with Stochastic Players that have the same confidence level and different participation probabilities, we show that they admit a lexicographic potential (Theorem 6.5), and thus a PNE, and also that a PNE can be computed by a simple greedy best response algorithm (Theorem 6.4), where the players sequentially adopt their best response strategy in non-decreasing order of participation probabilities, given the strategies of the previous players in the order. As for the inefficiency of PNE, we focus on parallel-link networks with linear latency functions, and prove that the Price of Anarchy (PoA) is $\Theta(n)$, where n is the number of players, in the case of stochastic players (Theorems 7.3 and 6.7), and may be unbounded, in the case of stochastic edges (Theorem 6.11).

2.4 Improving Selfish Routing through Risk Aversion

Motivated by the results of [71], we consider (nonatomic) routing games with risk-averse players and investigate how one can exploit risk-aversion and modify the perceived cost of the players so that the PoA wrt. the total latency of the players is significantly improved. To the best of our knowl-

edge, this is the first time that risk-aversion is proposed and investigated as a remedy to the inefficiency of selfish routing.

Our starting point is that in some practical applications, we may carefully introduce exogenous variance in the edge delays so that the expected delay does not change, but the risk-averse cost of the players increases. For example, in a transportation network, this can be done by randomly increasing or decreasing the proportion of time allocated to the green traffic light for short time intervals or by opening or closing an auxiliary traffic lane. In a telecommunications network, we might randomly increase or decrease the link capacity allocated to a particular type of traffic or change its priority. Thus, we assume that for each edge e, we can increase (by a small multiplicative factor) the delay through e with some positive probability p_e and also decrease it with some positive probability q_e, where p_e and q_e are typically small, so that the expected latency through e remains $d_e(x)$. On the other hand, the variance in the delay introduced by such random changes increases the perceived cost of risk-averse players. Therefore, by carefully adjusting the perceived cost of the players, we can control the network congestion, in a way conceptually similar to that of refundable tolls, and improve the PoA through a delicate and easy to implement mechanism that exploits the risk-aversion of the players.

More specifically, we assume that all the players are homogenous wrt. their risk attitude and that random changes on the edge delays cause the perceived delays of the players to increase by a (typically small) multiplicative factor[1]. In fact, we assume that the perceived delay function of each edge e changes from $d_e(x)$ to $(1 + \gamma_e)d_e(x)$, where $\gamma_e > 0$ may depend on the edge, the type and the probability of random changes, the risk attitude of the players, the exact formula of the risk-averse individual cost and most importantly, the practical setting. Although we briefly discuss, in Section 7.1, how γ_e is determined and provide some examples, for simplicity and generality, we deliberately avoid getting into the details of how γ_e's are precisely calculated. In contrast, we simply assume a given upper bound γ on the largest possible change in the delay functions and refer to the corresponding routing game as a γ-modifiable game (or instance), with the understanding in each particular application, γ can be determined by considering all the factors mentioned above. In this setting, a flow (and

[1]We mostly have in mind random (relatively small and short-term) changes that affect the link "capacity", e.g., opening or closing an auxiliary lane for a short time, increasing or decreasing the capacity of a telecommunication link by a certain factor, etc. As a result, we assume that the perceived cost of the players increases by a small multiplicative factor. This also distinguishes our technical approach and our results from the extensively studied case of additive refundable tolls (see e.g., [27, 35, 54, 55]).

in particular, the optimal flow, which minimizes the total actual latency of the players) is γ-*enforceable* (or simply, *enforceable*) if it is a Nash flow of the modified routing game with the perceived cost of each edge e equal to $(1+\gamma_e)d_e(x)$, for some $\gamma_e \in [0,\gamma]$.

On the technical side, we observe that the maximum ratio of the optimal marginal-cost toll to the optimal edge latency gives an upper bound on the value of γ required to enforce the optimal flow (Proposition 8). This establishes the applicability of our approach and formalizes its connection to refundable tolls. However, in addition to the fact that we use multiplicative latency modifications, an important difference of our approach from (additive) refundable tolls is that we always assume an upper bound γ on the change of the perceived latencies, while optimal tolls do not assume any bounds and can become arbitrarily large, e.g., as large as the delay on the maximum delay $s-t$ path used by the optimal flow (see e.g., [35, Theorem 4.1]).

Next, to demonstrate the efficiency of our approach, we focus on the simple and important special case of parallel-link networks (Section 7.2). We characterize the class of γ-modifiable routing games on parallel-links for which the optimal flow is enforceable (Theorem 7.1). Based on this characterization, we present a recursive procedure that given a γ-modifiable routing game and its optimal flow, computes a set of γ-bounded latency modifications and a γ-enforceable flow with PoA significantly less than the PoA of the original game (Lemma 7.2). Generalizing the variational inequality approach of [29], we prove that the PoA of the resulting flow is at most $\max\{1,(1-\beta_\gamma(\mathcal{D}))^{-1}\}$, where $\mathcal{D}$ is the class of the delay functions of the original routing game and

$$\beta_\gamma(\mathcal{D}) = \sup_{d\in\mathcal{D},x\geq y\geq 0} \frac{y(d(x)-d(y))-\gamma(x-y)d(x)}{xd(x)}$$

is a natural generalization of the quantity $\beta(\mathcal{D})$ introduced in [29] (Theorem 7.3). For example, our analysis implies that for linear delays, the PoA of the γ-enforceable flow computed by our approach is at most $\max\{1,(1-(1-\gamma)^2/4)^{-1}\}$ (Corollary 7.5), which is significantly less that $4/3$ even for small values of γ (e.g., it is less than $6/5$ for $\gamma = 0.1$). We also show that our PoA analysis in terms of γ is tight wrt. all γ-enforceable flows (Theorem 7.7). Finally, we give a procedure that computes a set of γ-bounded latency modifications and a γ-enforceable flow with such PoA in time polynomially related to the time needed for computing Nash flows in parallel-links (Lemma 7.8). Therefore, given any γ-modifiable game on parallel-links, we can efficiently (for a wide class of latency functions) compute a game with perceived delays changed by a factor of at most $1+\gamma$

such that the PoA is at most $\max\{1, (1-\beta_\gamma(\mathcal{D}))^{-1}\}$.

In Section 7.3 we relate γ-modifiable CGs to routing games with restricted tolls ([17]). We point out the results of [17] that can be applied to γ-modifiable CGs while we support our constructive and more complicated approach for providing better insight for the problem of finding optimal γ-modifications and for the improvement on the PoA.

In section 7.4, we discuss how our approach can be extended to more general settings. First we give the intuition of how similar to the parallel links case results can be derived for the case of series parallel networks. Since we have completed the proofs that our results can be extended to the case of series parallel networks only recently, we chose not to include these results in this thesis. Instead, we explain the approach and provide sketches of the main ideas. Right after we discuss how we can derive similar results for the case of parallel links with heterogeneous players and more general restrictions on the uncertainty that is allowed to be added. Note here that this case is not solved by none of [51], [17] and [53], that deal with CGs with bounds on the allowable tolls.

Chapter 3

Congestion Games Preliminaries

In this chapter we state basic definitions needed in the following chapters. Model specific definitions and properties are given separately at the beginning of each chapter.

3.1 General Notation and Conventions

For a random variable X, $\mathbf{E}[X]$ denotes the **expectation** of X and $\mathbf{Var}[X]$ denote the *variance* of X. For an event E in a sample space, $\mathbf{Pr}[E]$ denotes the probability of E happening. We say that an event E occurs **with high probability**, if there is a constant $a \geq 1$, such that $\mathbf{Pr}[E] \geq 1 - n^{-a}$, where n usually denotes the number of vertices of the network G to which E refers. We implicitly use the union bound to account for the occurrence of more than one low probability events.

For any integer $n \geq 1$, we let $[n] = \{1, \ldots, n\}$

A latency function $c_e(x)$ is **linear** if $c_e(x) = a_e x$, for some $a_e > 0$, and **affine** if $c_e(x) = a_e x + b_e$, for some $a_e, b_e \geq 0$. We say that a latency function $c_e(x)$ satisfies the **Lipschitz condition** with constant $\xi > 0$, if for all $x, y \in [0, r]$, $|c_e(x) - c_e(y)| \leq \xi |x - y|$.

3.2 Congestion Games Definitions

An **atomic Congestion Game** is a tuple $\mathcal{G}(N, E, (S_i)_{i \in N}, (d_e)_{e \in E})$, where N denotes the set of players, E denotes the set of resources, $S_i \subseteq 2^E \setminus \{\emptyset\}$ denotes the strategy space of each player i and $d_e : \mathbb{N} \mapsto \mathbb{R}_{\geq 0}$ is a non-negative and non-decreasing latency function associated with each resource e.

Network Congestion Games are CGs in which players have a source node, s_i, and a target node, t_i, on a directed graph (network) and the strategy space of each player i is consisted by the paths that connect s_i with t_i. The resources of the CG are exactly the edges of the network each of which comes together with a latency (cost) function.

An **atomic Network CG** is a tuple $\mathcal{G}(G(V,E),(d_e)_{e\in E},K,\{N\}_{i=1\ldots|K|})$, where $G=(V,E)$ denotes a directed network, $K\subseteq V\times V$ is a set of s_i-t_i source target pairs, $d_e:\mathbb{R}_{\geq 0}\mapsto\mathbb{R}_{\geq 0}$ is a non-negative and non-decreasing latency function associated with each edge e and $N=(n_1,\ldots,n_{|K|})$ is the vector of players, where n_i is the number of players that must move from source s_i to target t_i.

Non atomic network CGs are like atomic network CGs with infinite, infinitesimal players, i.e. the presence or the absence of a player on a resource does not affect the congestion on the resource. The infinite players that share the same s_i-t_i pair form an amount of flow that must be routed from s_i to t_i.

A **non atomic Network CG** is a tuple $\mathcal{G}(G(V,E),(d_e)_{e\in E},K,r)$, where $G=(V,E)$ denotes a directed network, $K\subseteq V\times V$ is a set of s_i-t_i source target pairs, $d_e:\mathbb{R}_{\geq 0}\mapsto\mathbb{R}_{\geq 0}$ is a non-negative and non-decreasing latency function associated with each edge e and $r=(r_1,\ldots,r_{|K|})$ is a vector of flows that must be routed, where r_i is the amount of flow that must be routed from source i to target i.

Network CGs are also referred as **Selfish Routing Games**. In a CG, if all players share the same strategy space then we have a *symmetric* CG. We deal with symmetric network CGs case where K contains a single $s-t$ pair and thus, assuming that s and t are specified in G, we use the terminology $\mathcal{G}(G(V,E),(d_e)_{e\in E},n)$ for atomic network CGs, where n is the number of players that use the network, and $\mathcal{G}(G(V,E),(d_e)_{e\in E},r)$ for non atomic network CGs, where r is an amount of flow to be routed in the network. We let $\mathcal{P}$ be the set of paths connecting s to t. We may also use $\mathcal{G}(G(V,E),n)$ or $\mathcal{G}(G(V,E),r)$ respectiely if the set of latency functions under use is clear from the context or is assumed to be given within the network G.

Subnetworks and Subinstances. Given an instance $\mathcal{G}=(G(V,E),(d_e)_{e\in E},x)$, any subgraph $H(V,E')$, $E'\subseteq E$, obtained from G by edge deletions, is a *subnetwork* of G. H has the same origin s and destination t as G, and the edges of H have the same latency functions as in $\mathcal{G}$. Each instance $\mathcal{H}=(H(V,E'),(d_e)_{e\in E'},x)$, where $H(V,E')$ is a subnetwork of $G(V,E)$, is a *subinstance* of $\mathcal{G}$.

In atomic CGs, a **configuration** is a vector $s=(s_1,\ldots,s_n)$ consisting of

a strategy $s_i \in S_i$ for each player i. We let $s_e = |\{i : e \in s_i\}|$ denote the congestion induced on each resource e by s.

In non atomic network CGs, a (feasible) **flow** f is a non-negative vector indexed by $\mathcal{P}$ so that $\sum_{p\in\mathcal{P}} f_p = r$. For a flow f and each edge e, we let $f_e = \sum_{p:e\in p} f_p$ denote the amount of flow that f routes through e. An edge e is used by flow f if $f_e > 0$. A path p is used by flow f if all its edges are used, i.e. $\min_{e\in p}\{f_e\} > 0$. Somehow abusing notation, if p is used we may write $f_p > 0$.

Additive Costs. In additive costs atomic CGs, the cost of the strategy s_i of player i under configuration s is $d_i(s) = \sum_{e\in s_i} d_e(s_e)$. Similar, for additive costs non atomic network CGs, the cost of a path p under flow f is $d_p(f) = \sum_{e\in p} d_e(s_e)$

Bottleneck Costs. In bottleneck costs atomic CGs, the cost of the strategy s_i of player i under configuration s is $b_i(s) = \max_{e\in s_i} d_e(s_e)$. Similar, for bottleneck costs non atomic network CGs, the cost of a path p under flow f is $b_p(f) = \max_{e\in p} d_e(s_e)$.

3.3 Equilibria

There are different ways to define an equilibrium. We are concerned in the case of *Pure Nash Equilibrium*. Informally, a configuration s or a flow f is a *Pure Nash equilibrium* if no player can improve her individual cost by unilaterally changing her strategy.

Pure Nash Equilibrium - **Nash flows**. For an atomic additive costs CG, a configuration s is said to be a Pure Nash Equilibrium (PNE) if $d_i(s_i, s_{-i}) \leq d_i(s'_i, s_{-i}), \forall s'_i \in S_i$, where $d_i(k, s_{-i})$ denotes the cost of player i for the configuration where i plays strategy k and the rest of the players play according to configuration s. For a non atomic additive costs network CG, f is a Nash Equilibrium or a Nash flow if for all $s-t$ paths p, p', if $f_p > 0$, then $d_p(f) \leq d_{p'}(f)$.

Similar, for an atomic bottleneck costs CG, a configuration s is said to be a Pure Nash Equilibrium if $b_i(s_i, s_{-i}) \leq b_i(s'_i, s_{-i}), \forall s'_i \in S_i$, where $b_i(k, s_{-i})$ denotes the cost of player i for the configuration where i plays strategy k and the rest of the players play according to configuration s. For a non atomic bottleneck costs network CG, f is a Nash Equilibrium or a Nash flow if for all $s-t$ paths p, p', if $f_p > 0$, then $b_p(f) \leq b_{p'}(f)$. A basic property of non atomic network CGs is that all players incur the same latency either under additive or under bottleneck costs.

ε-Nash Equlibria (Flows). The definition of a Nash Equilibrium can be

generalized to that of an "almost Nash" Equilibrium: For some constant $\varepsilon > 0$, a configuration or flow x is an ε-Nash Equilibrium if for all $s - t$ paths p, p', if $x_p > 0$, $d_p(x) \leq d_{p'}(x) + \varepsilon$, for additive costs games and $b_p(x) \leq b_{p'}(x) + \varepsilon$, for bottleneck costs games.

Potential Functions. Games do not always possess PNE, though additive costs CGs always posses one. This can be proved via a potential function method. A function $\Phi : S_1 \times \cdots \times S_n \mapsto \mathbb{R}_{\geq 0}$ is an *ordinal potential function* for a CG if $d_i(s_{-i}, s'_i) - d_i(s) < 0 \Leftrightarrow \Phi(s_{-i}, s'_i) - \Phi(s) < 0$

Intuitively an ordinal potential is a function that follows the sign of change of any players cost when she changes unilaterally. The admittance of the above type of potential is a necessary and sufficient condition for a game to have the Finite Improvement Property, i.e. every best response sequence is finite. So if letting the players, one at a time, to change their strategy to their best response then, at the end, we get a configuration that is a PNE.

A more strong kind of potential is the one encountered so far in CGs where a function tracks not only the sign but also the amount of the change. A function $\Phi : S_1 \times \cdots \times S_n \mapsto \mathbb{R}_{\geq 0}$ is an *exact potential* for a CG if $\Phi(s_{-i}, s'_i) - \Phi(s) = d_i(s_{-i}, s'_i) - d_i(s)$

Games that admit an exact potential are called *Potential Games*. It can be proved that additive costs CGs are isomorphic to Potential games. One direction is easy: CGs (atomic or non atomic with additive costs) always admit an exact potential:

atomic case

$$\Phi(s) = \sum_{e \in E} \sum_{i=1}^{s_e} d_e(i)$$

non atomic case

$$\Phi(s) = \sum_{e \in E} \int_0^{s_e} d_e(x)dx$$

For CGs with bottleneck costs, best response dynamics can also be used to prove the existence of (optimal) equilibrium, e.g. [11, Corollary 2], although, directly, the optimal solution (with respect to the social cost function defined in the next section) is a Nash equilibrium.

3.4 Price of Anarchy and Price of Stability

Players' strategies choices cause an overall charge on the resources of the game that can be seen as an overall cost charging the network manager or

the players' "society". This is quantified via a *Social Cost Function*.

A **Social Cost Function**, for atomic games, is a function $SC(s) : S_1 \times \ldots \times S_n \to \mathbb{R}_{\geq 0}$. The most popular Social Cost Function used in additive costs CGs is the sum of the players' costs (or, somehow equivalently, the average of the players' costs), i.e. $SC(s) = \sum_i d_i(s_i)$. For bottleneck costs games, the most used social cost function is $SC(s) = \max_i b_i(s_i) = \max_{e:s_e>0} d_e(s_e)$. Similar, for non atomic network CGs it is $SC(f) : \textit{Space of Flows} \to \mathbb{R}_{\geq 0}$ and for additive costs CGs, $SC(f) = \sum_p f_p d_p(f_p)$ while for bottleneck costs CGs, $SC(f) = \max_{p:f_p>0} b_p(f_p) = \max_{e:f_e>0} d_e(s_e)$. For ease of notation, for bottleneck CGs, we may use $B(s)$ and $B(f)$ instead of $SC(s)$ and $SC(f)$ respectively.

Optimal Solutions. A configuration that minimizes the social cost function is an optimum configuration called OPT. Similar, the optimal flow is defined as the flow that minimizes the social cost function. PNE are considered as the possible outcomes of the game, yet they do not necessarily minimize social cost. This causes inefficiency to the network. To capture this inefficiency there are two measures, one adopting a worst case approach (PoA) and the other a best case approach (PoS).

Price of Anarchy (PoA) and Price of Stability (PoS). By letting x denote a configuration for the case of atomic CGs and a flow for the case of non atomic CGs, define

$$PoA = max\Big\{\frac{SC(x)}{SC(OPT)} | x \textit{ is a PNE}\Big\} \qquad PoS = min\Big\{\frac{SC(x)}{SC(OPT)} | x \textit{ is a PNE}\Big\}$$

The PoA of symmetric non atomic additive costs CGs is independent of the structure of the network as it was first shown in [74]. Correa et al. in [28] prove a general tight bound for the PoA based only on the class $\mathcal{D}$ of latency functions, which is $\rho(\mathcal{D}) = (1 - \beta(\mathcal{D}))^{-1}$, where $\beta(\mathcal{D}) = \sup_{d\in\mathcal{D}, x\geq y\geq 0} \frac{y(d(x)-d(y))}{xd(x)}$. The same bound holds for the *PoS* in symmetric atomic network CGs and for the *PoA* on games on extension parallel networks as shown in [39]. See also e.g. [22] and [79] for similar results for atomic CGs.

For bottleneck network CGs, although the PoS is equal to 1 (see e.g. [11, Corollary 2]) the PoA behaves way much worse as simple examples give a tight bound of $\Omega(|V|)$ (see figure 8.2 or e.g. [27, Figure 2]), where $|V|$ is the number of vertices of the network.

Chapter 4

On the Hardness of Network Design for Bottleneck Routing Games

In this chapter, we investigate the computational complexity and the approximability of the network design problem for non-atomic bottleneck routing games, where the individual cost of each player is the bottleneck cost of her path, and the social cost is the bottleneck cost of the network, i.e. the maximum latency of a used edge.

We first show that bottleneck routing games do not suffer from Braess's paradox either if the network is series-parallel, or if we consider only subpath-optimal Nash flows, a subclass of Nash flows (formally defined in section 4.1).

On the negative side, we prove that even for games with strictly increasing linear latencies, it is NP-hard not only to recognize instances suffering from the paradox, but also to distinguish between instances for which the Price of Anarchy (PoA) can decrease to 1 and instances for which the PoA cannot be improved by edge removal, even if their PoA is as large as $\Omega(n^{0.121})$. This implies that the network design problem for linear bottleneck routing games is NP-hard to approximate within a factor of $O(n^{0.121-\varepsilon})$, for any constant $\varepsilon > 0$. The proof is based on a recursive construction of hard instances that carefully exploits the properties of bottleneck routing games, and may be of independent interest.

On the positive side, we present an algorithm for finding a subnetwork that is almost optimal w.r.t. the bottleneck cost of its worst Nash flow, when the worst Nash flow in the best subnetwork routes a non-negligible amount of flow on all used edges. We show that the running time is essentially determined by the total number of paths in the network, and is quasipolynomial when the number of paths is quasipolynomial.

4.1 Problem-Specific Definitions and Facts

We deal with a typical instance of a non-atomic *bottleneck routing game*, $\mathcal{G} = (G(V,E), (d_e)_{e\in E}, r)$.

Optimal and Nash Flow Properties. Let o denote the optimal flow of an instance $\mathcal{G}$. We let $B^*(\mathcal{G}) = B(o)$. We note that for every subinstance $\mathcal{H}$ of $\mathcal{G}$, $B^*(\mathcal{H}) \geq B^*(\mathcal{G})$.

As noted earlier, in a Nash flow f all players incur a common bottleneck cost, i.e. $B(f) = \min_p b_p(f)$, and for every $s-t$ path p', $B(f) \leq b'_p(f)$. We observe that if a flow f is a Nash flow for an $s-t$ network $G(V,E)$, then the set of edges e with $d_e(f_e) \geq B(f)$ comprises an $s-t$ cut in G. For the converse, if for some flow f, there is an $s-t$ cut consisting of edges e either with $f_e > 0$ and $d_e(f_e) = B(f)$, or with $f_e = 0$ and $d_e(f_e) \geq B(f)$, then f is a Nash flow. Moreover, for all bottleneck routing games with linear latencies $a_e x$, a flow f is a Nash flow iff the set of edges e with $d_e(f_e) = B(f)$ comprises an $s-t$ cut.

It can be shown that every bottleneck routing game admits at least one Nash flow (see e.g., [27, Proposition 2]), and that there is an optimal flow that is also a Nash flow (see e.g., [11, Corollary 2]). In general, a bottleneck routing game admits many different Nash flows, each with a possibly different bottleneck cost of the players. Given an instance $\mathcal{G}$, we let $B(\mathcal{G})$ denote the bottleneck cost of the players in the worst Nash flow of $\mathcal{G}$, i.e. the Nash flow f that maximizes $B(f)$ among all Nash flows. We refer to $B(\mathcal{G})$ as the worst equilibrium bottleneck cost of $\mathcal{G}$. For convenience, for an instance $\mathcal{G} = (G, c, r)$, we sometimes write $B(G, r)$, instead of $B(\mathcal{G})$, to denote the worst equilibrium bottleneck cost of $\mathcal{G}$. We note that for every subinstance $\mathcal{H}$ of $\mathcal{G}$, $B^*(\mathcal{G}) \leq B(\mathcal{H})$, and that there may be subinstances $\mathcal{H}$ with $B(\mathcal{H}) < B(\mathcal{G})$, which is the essence of Braess's paradox (see e.g., Fig. 5.1).

The following proposition considers the effect of a uniform scaling of the latency functions.

Proposition 1. *Let $\mathcal{G} = (G, c, r)$ be a routing instance, let $a > 0$, and let $\mathcal{G}' = (G, ac, r)$ be the routing instance obtained from $\mathcal{G}$ if we replace the latency function $d_e(x)$ of each edge e with $ad_e(x)$. Then, any $\mathcal{G}$-feasible flow f is also $\mathcal{G}'$-feasible and has $B_{\mathcal{G}'}(f) = aB_{\mathcal{G}}(f)$. Moreover, a flow f is a Nash flow (resp. optimal flow) of $\mathcal{G}$ iff f is a Nash flow (resp. optimal flow) of $\mathcal{G}'$.*

Proof. Since the traffic rate of both $\mathcal{G}$ and $\mathcal{G}'$ is r, any $\mathcal{G}$-feasible flow f is also $\mathcal{G}'$-feasible. Moreover, the $\mathcal{G}'$-latency of f on each edge e is $ad_e(f_e)$. This immediately implies that $B_{\mathcal{G}'}(f) = aB_{\mathcal{G}}(f)$, and that f is a Nash flow (resp. optimal flow) of $\mathcal{G}$ iff f is a Nash flow (resp. optimal flow) of $\mathcal{G}'$. □

Subpath-Optimal Nash Flows. For a flow f and any vertex u, let $b_f(u)$ denote the minimum bottleneck cost of f among all $s-u$ paths. The flow f is a *subpath-optimal Nash flow* [27] if for any vertex u and any $s-t$ path p with $f_p > 0$ that includes u, the bottleneck cost of the $s-u$ part of p is $b_f(u)$. For example, the Nash flow f in Fig. 5.1.a is not subpath-optimal, because $b_f(v) = 0$, through the edge (s, v), while the bottleneck cost of the path (s, u, v) is 1. For this instance, the only subpath-optimal Nash flow is the optimal flow with $1/2$ unit on the path (s, u, t) and $1/2$ unit on the path (s, v, t).

We formally define the problems we will see

Problem Definitions.

- **Paradox-Ridden Recognition** (ParRidBC): Given an instance $\mathcal{G}$, decide if $\mathcal{G}$ is paradox-ridden.
- **Best Subnetwork** (BSubNBC): Given an instance $\mathcal{G}$, find the best subnetwork H^* of G.

We investigate the complexity and the approximability of these fundamental selfish network design problems for bottleneck routing games.

We note that the objective function of BSubNBC is the worst equilibrium bottleneck cost $B(H, r)$ of a subnetwork H. Thus, a (polynomial-time) algorithm A achieves an a-approximation for BSubNBC if for all instances $\mathcal{G}$, A returns a subnetwork H with $B(H, r) \leq aB(H^*, r)$. A subtle point is that given a subnetwork H, we do not know how to efficiently compute the worst equilibrium bottleneck cost $B(H, r)$ (see also [10, 52], where a similar issue arises). To deal with this delicate issue, our hardness results use a certain subnetwork structure to identify a good approximation to BSubNBC.

Series-Parallel Networks. A directed $s-t$ network is *series-parallel* if it either consists of a single edge (s, t) or can be obtained from two series-parallel graphs with terminals (s_1, t_1) and (s_2, t_2) composed either in series or in parallel. In a *series composition*, t_1 is identified with s_2, s_1 becomes s, and t_2 becomes t. In a *parallel composition*, s_1 is identified with s_2 and becomes s, and t_1 is identified with t_2 and becomes t.

4.2 Paradox-Free Network Topologies and Paradox-Free Nash Flows

We start by discussing two interesting cases where Braess's paradox does not occur. We first show that if we have a bottleneck routing game $\mathcal{G}$ de-

fined on an $s-t$ series-parallel network, then $\rho(\mathcal{G}) = 1$, and thus Braess's paradox does not occur. We recall that this was also pointed out in [31] for the case of atomic unsplittable bottleneck routing games. Moreover, we note that a directed $s-t$ network is series-parallel iff it does not contain a ∂-graph with degree-2 terminals as a topological minor. Therefore, the example in Fig. 5.1 demonstrates that series-parallel networks is the largest class of network topologies for which Braess's paradox does not occur (see also [64] for a similar result for the case of additive routing games).

Proposition 2. *Let $\mathcal{G}$ be bottleneck routing game on an $s-t$ series-parallel network. Then, $\rho(\mathcal{G}) = 1$.*

Proof. Let f be any Nash flow of $\mathcal{G}$. We use induction on the series-parallel structure of the network G, and show that f is an optimal flow w.r.t the bottleneck cost, i.e., that $B(f) = B^*(\mathcal{G})$. For the basis, we observe that the claim holds if G consists of a single edge (s,t). For the inductive step, we distinguish two cases, depending on whether G is obtained by the series or the parallel composition of two series-parallel networks G_1 and G_2.

Series Composition. First, we consider the case where G is obtained by the series composition of an $s-t'$ series-parallel network G_1 and a $t'-t$ series-parallel network G_2. We let f_1 and f_2, both of rate r, be the restrictions of f into G_1 and G_2, respectively.

We start with the case where $B(f) = B(f_1) = B(f_2)$. Then, either f_1 is a Nash flow in G_1, or f_2 is a Nash flow in G_2. Otherwise, there would be a $s-t'$ path p_1 in G_1 with bottleneck cost $b_{p_1}(f_1) < B(f_1)$, and an $t'-t$ path p_2 in G_2, with bottleneck cost $b_{p_2}(f_2) < B(f_2)$. Combining p_1 and p_2, we obtain an $s-t$ path $p = p_1 \cup p_2$ in G with bottleneck cost smaller than $B(f)$, which contradicts the hypothesis that f is a Nash flow of $\mathcal{G}$. If f_1 (or f_2) is a Nash flow in G_1 (resp. G_2), then by induction hypothesis f_1 (resp. f_2) is an optimal flow in G_1 (resp. in G_2), and thus f is an optimal flow of $\mathcal{G}$.

Otherwise, we assume, without loss of generality, that $B(f) = B(f_1) < B(f_2)$. Then, f_1 is a Nash flow in G_1. Otherwise, there would be an $s-t'$ path p_1 in G_1 with bottleneck cost $b_{p_1}(f_1) < B(f_1)$, which could be combined with any $t'-t$ path p_2 in G_2, with bottleneck cost $B(f_2) < B(f_1)$, into an $s-t$ path $p = p_1 \cup p_2$ with bottleneck cost smaller than $B(f)$. The existence of such a path p contradicts the the hypothesis that f is a Nash flow of $\mathcal{G}$. Therefore, by induction hypothesis f_1 is an optimal flow in G_1, and thus f is an optimal flow of $\mathcal{G}$.

Parallel Composition. Next, we consider the case where G is obtained by the parallel composition of an $s-t$ series-parallel network G_1 and an $s-t$ series-parallel network G_2. We let f_1 and f_2 be the restriction of f into

G_1 and G_2, respectively, let r_1 (resp. r_2) be the rate of f_1 (resp. f_2), and let $\mathcal{G}_1$ (resp. $\mathcal{G}_2$) be the corresponding routing instance. Then, since f is a Nash flow of $\mathcal{G}$, f_1 and f_2 are Nash flows of $\mathcal{G}_1$ and $\mathcal{G}_2$ respectively, and $B(f_1) = B(f_2) = B(f)$. Therefore, by the induction hypothesis, f_1 and f_2 are optimal flows of $\mathcal{G}_1$ and $\mathcal{G}_2$, and f is an optimal flow of $\mathcal{G}$. To see this, we observe that any flow different from f must route more flow through either G_1 or G_2. But if the flow through e.g. G_1 is more than r_1, the bottleneck cost through G_1 would be at least as large as $B(f_1)$. □

Next, we show that any subpath-optimal Nash flow achieves a minimum bottleneck cost, and thus Braess's paradox does not occur if we restrict ourselves to subpath-optimal Nash flows.

Proposition 3. *Let $\mathcal{G}$ be bottleneck routing game, and let f be any subpath-optimal Nash flow of $\mathcal{G}$. Then, $B(f) = B^*(\mathcal{G})$.*

Proof. Let f be any subpath-optimal Nash flow of $\mathcal{G}$, let S be the set of vertices reachable from s via edges with bottleneck cost less than $B(f)$, let $\delta^+(S)$ be the set of edges $e = (u, v)$ with $u \in S$ and $v \notin S$, and let $\delta^-(S)$ be the set of edges $e = (u, v)$, with $u \notin S$ and $v \in S$. Then, in [27, Lemma 4.5], it is shown that (i) $(S, V \setminus S)$ is an $s - t$ cut, (ii) for all edges $e \in \delta^+(S)$, $d_e(f_e) \geq B(f)$, (iii) for all edges $e \in \delta^+(S)$ with $f_e > 0$, $d_e(f_e) = B(f)$, and (iv) for all edges $e \in \delta^-(S)$, $f_e = 0$.

By (i) and (iv), any optimal flow o routes at least as much traffic as the subpath-optimal Nash flow f routes through the edges in $\delta^+(S)$. Thus, there is some edge $e \in \delta^+(S)$ with $o_e \geq f_e$, which implies that $d_e(o_e) \geq d_e(f_e) \geq B(f)$, where the second inequality follows from (ii). Since $B^*(\mathcal{G}) = B(o) \geq d_e(o_e)$, we obtain that $B^*(\mathcal{G}) = B(f)$. □

4.3 Recognizing Paradox-Ridden Instances is Hard

In this section, we show that given a linear bottleneck routing game $\mathcal{G}$, it is NP-hard not only to decide whether $\mathcal{G}$ is paradox-ridden, but also to approximate the best subnetwork within a factor less than 4/3. To this end, we employ a reduction from the 2-Directed Disjoint Paths problem (2-DDP), where we are given a directed network D and distinguished vertices s_1, s_2, t_1, t_2, and ask whether D contains a pair of vertex-disjoint paths connecting s_1 to t_1 and s_2 to t_2. 2-DDP was shown NP-complete in [37, Theorem 3], even if the network D is known to contain two edge-disjoint paths connecting s_1 to t_2 and s_2 to t_1. In the following, we say that a subnetwork D' of D is *good* if D' contains (i) at least one path outgoing from each of s_1

and s_2 to either t_1 or t_2, (ii) at least one path incoming to each of t_1 and t_2 from either s_1 or s_2, and (iii) either no $s_1 - t_2$ paths or no $s_2 - t_1$ paths. We say that D' is *bad* if any of these conditions is violated by D'. We note that we can efficiently check whether a subnetwork D' of D is good, and that a good subnetwork D' serves as a certificate that D is a YES-instance of 2-DDP. Then, the following lemma directly implies the hardness result of this section.

Lemma 4.1. Let $\mathcal{I} = (D, s_1, s_2, t_1, t_2)$ be any 2-DDP instance. Then, we can construct, in polynomial time, an $s-t$ network $G(V, E)$ with a linear latency function $d_e(x) = a_e x$, $a_e > 0$, on each edge e, so that for any traffic rate $r > 0$, the bottleneck routing game $\mathcal{G} = (G, c, r)$ has $B^*(\mathcal{G}) = r/4$, and:

1. If $\mathcal{I}$ is a YES-instance of 2-DDP, there exists a subnetwork H of G with $B(H, r) = r/4$.
2. If $\mathcal{I}$ is a NO-instance of 2-DDP, for all subnetworks H' of G, $B(H', r) \geq r/3$.
3. For all subnetworks H' of G, either H' contains a good subnetwork of D, or $B(H', r) \geq r/3$.

Proof. We construct a network $G(V, E)$ with the desired properties by adding 4 vertices, s, t, v, u, to D and 9 "external" edges $e_1 = (s, u)$, $e_2 = (u, v)$, $e_3 = (v, t)$, $e_4 = (s, v)$, $e_5 = (v, s_1)$, $e_6 = (s, s_2)$, $e_7 = (t_1, u)$, $e_8 = (u, t)$, $e_9 = (t_2, t)$ (see also Fig. 4.1.a). The external edges e_1 and e_3 have latency $d_{e_1}(x) = d_{e_3}(x) = x/2$. The external edges $e_4, \ldots, e_9$ have latency $d_{e_i} = x$. The external edge e_2 and each edge e of D have latency $d_{e_2}(x) = d_e(x) = \varepsilon x$, for some $\varepsilon \in (0, 1/4)$.

We first show that $B^*(\mathcal{G}) = r/4$. As for the lower bound, since the edges e_1, e_4, and e_6 form an $s - t$ cut in G, every $\mathcal{G}$-feasible flow has a bottleneck cost of at least $r/4$. As for the upper bound, we may assume that D contains an $s_1 - t_2$ path p and an $s_2 - t_1$ path q, which are edge-disjoint (see also [37, Theorem 3]). Then, we route a flow of $r/4$ through each of the paths (e_4, e_5, p, e_9) and (e_6, q, e_7, e_8), and a flow of $r/2$ through the path (e_1, e_2, e_3), which gives a bottleneck cost of $r/4$.

Next, we show (1), namely that if $\mathcal{I}$ is a YES-instance of 2-DDP, then there exists a subnetwork H of G with $B(H, r) = r/4$. By hypothesis, there is a pair of vertex-disjoint paths in D, p and q, connecting s_1 to t_1, and s_2 to t_2. Let H be the subnetwork of G that includes all external edges and only the edges of p and q from D (see also Fig. 4.1.b). We let $\mathcal{H} = (H, c, r)$ be the corresponding subinstance of $\mathcal{G}$. The flow routing $r/4$ units through

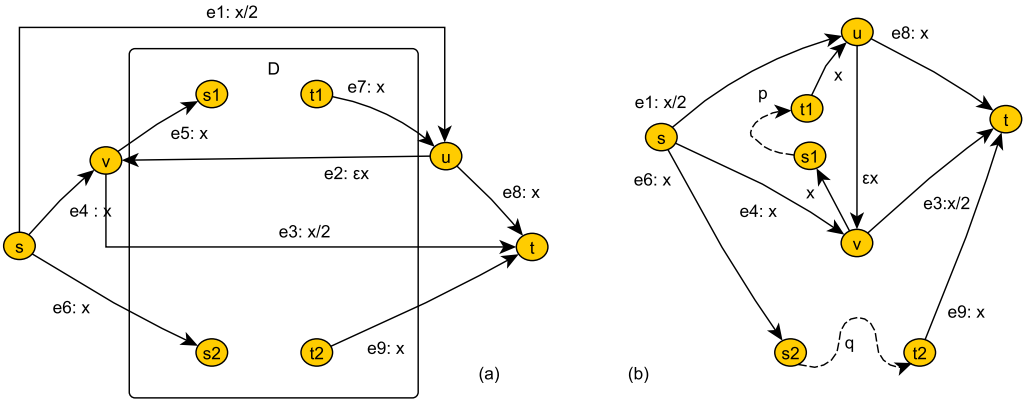

Figure 4.1: (a) The network G constructed in the proof of Lemma 4.1. (b) The best subnetwork of G, with PoA = 1, for the case where D contains a pair of vertex-disjoint paths connecting s_1 to t_1 and s_2 to t_2.

each of the paths (e_4, e_5, p, e_7, e_8) and (e_6, q, e_9), and $r/2$ units through the path (e_1, e_2, e_3), is an $\mathcal{H}$-feasible Nash flow with a bottleneck cost of $r/4$.

We proceed to show that any Nash flow of $\mathcal{H}$ achieves a bottleneck cost of $r/4$. For sake of contradiction, let f be a Nash flow of $\mathcal{H}$ with $B(f) > r/4$. Since f is a Nash flow, the edges e with $d_e(f_e) \geq B(f)$ form an $s-t$ cut in H. Since the bottleneck cost of e_2 and of any edge in p and q is at most $r/4$, this cut includes either e_6 or e_9 (or both), either e_1 or e_3 (or both), and either e_4 or e_8 (or e_5 or e_6, in certain combinations with other edges). Let us consider the case where this cut includes e_1, e_4, and e_6. Since the bottleneck cost of these edges is greater than $r/4$, we have more than $r/2$ units of flow through e_1 and more than $r/4$ units of flow through each of e_4 and e_6. Hence, we obtain that more than r units of flow leave s, a contradiction. All other cases are similar.

To conclude the proof, we have also to show (3), namely that for any subnetwork H' of G, if H' does not contain a good subnetwork of D, then $B(H', r) \geq r/3$. We observe that (3) implies (2), because if $\mathcal{I}$ is a NO-instance, any two paths, p and q, connecting s_1 to t_1 and s_2 to t_2, have some vertex in common, and thus, D includes no good subnetworks. To show (3), we let H' be any subnetwork of G, and let $\mathcal{H}'$ be the corresponding subinstance of $\mathcal{G}$. We first show that either H' contains (i) all external edges, (ii) at least one path outgoing from each of s_1 and s_2 to either t_1 or t_2, and (iii) at least one path incoming to each of t_1 and t_2 from either s_1 or s_2, or H' includes a "small" $s-t$ cut, and thus any $\mathcal{H}'$-feasible flow f has $B(f) \geq r/3$.

To prove (i), we observe that if some of the edges e_1, e_4, and e_6 is missing from H', r units of flow are routed through the remaining ones,

which results in a bottleneck cost of at least $r/3$. The same argument applies to the edges e_3, e_8, and e_9. Similarly, if e_2 is not present in H', the edges e_4, e_6, and e_8 form an $s-t$ cut, and routing r units of flow through them causes a bottleneck cost of at least $r/3$. Therefore, we can assume, without loss of generality, that all these external edges are present in H'.

Now, let us focus on the external edges e_5 and e_7. If e_5 is not present in H' and there is a path p outgoing from s_2 to either t_1 or t_2, routing $2r/3$ units of flow through the path (e_1, e_2, e_3) and $r/3$ units through the path (e_6, p, e_9) (or through the path (e_6, p, e_7, e_8)) is a Nash flow with a bottleneck cost of $r/3$ (see also Fig. 4.2.a). If s_2 is connected to neither t_1 nor t_2 (no matter whether e_5 is present in H' or not), the edges e_1 and e_4 form an $s-t$ cut, and thus, any $\mathcal{H}'$-feasible flow has a bottleneck cost of at least $r/3$. Similarly, we can show that if either e_7 is not present in H', or neither s_1 nor s_2 is connected to t_2, any $\mathcal{H}'$-feasible flow has a bottleneck cost of at least $r/3$. Therefore, we can assume, without loss of generality, that all external edges are present in H', and that H' includes at least one path outgoing from s_2 to either t_1 or t_2, and at least one path incoming to t_2 from either s_1 or s_2.

Similarly, we can assume, without loss of generality, that H' includes at least one path outgoing from s_1 to either t_1 or t_2, and at least one path incoming to t_1 from either s_1 or s_2. E.g., if s_1 is connected to neither t_1 nor t_2, routing $2r/3$ units of flow through the path (e_1, e_2, e_3) and $r/3$ units through s_2 and either t_1 or t_2 (or both) is a Nash flow with a bottleneck cost of $r/3$. A similar argument applies to the case where neither s_1 nor s_2 is connected to t_1.

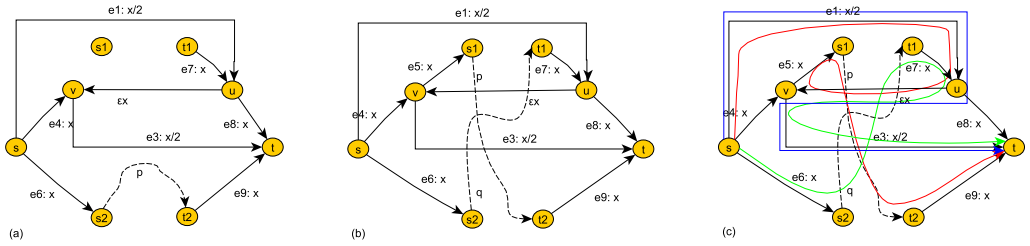

Figure 4.2: Possible subnetworks of G when there is no pair of vertex-disjoint paths connecting s_1 to t_1 and s_2 to t_2. The subnetwork (a) contains an s_2-t_2 path and does not include e_5. In the subnetwork (b), we essentially have all edges of G. In (c), we depict a Nash flow that consists of three paths, each carrying $r/3$ units of flow, and has a bottleneck cost of $r/3$.

Let us now consider a subnetwork H' of G that does not contain a good

subnetwork of D, but it contains (i) all external edges, (ii) at least one path outgoing from each of s_1 and s_2 to either t_1 or t_2, and (iii) at least one path incoming to each of t_1 and t_2 from either s_1 or s_2. By (ii) and (iii), and the hypothesis that the subnetwork of D included in H' is bad, H' contains an $s_1 - t_2$ path p and an $s_2 - t_1$ path q (see also Fig. 4.2.b). At the intuitive level, this corresponds to the case where no edges are removed from G. Then, routing $r/3$ units of flow on each of the $s-t$ paths (e_1, e_2, e_3), (e_1, e_2, e_5, p, e_9), and (e_6, q, e_7, e_2, e_3) has a bottleneck cost of $r/3$ and is a Nash flow, because the set of edges with bottleneck cost $r/3$ comprises an $s - t$ cut (see also Fig. 4.2.c). Therefore, we have shown part (3) of the lemma, which in turn, immediately implies part (2). □

We note that the bottleneck routing game $\mathcal{G}$ in the proof of Lemma 4.1 has $\rho(\mathcal{G}) = 4/3$, and is paradox-ridden, if $\mathcal{I}$ is a YES instance of 2-DDP, and paradox-free, otherwise. Thus, we obtain that:

Theorem 4.2. Deciding whether a bottleneck routing game with strictly increasing linear latencies is paradox-ridden is NP-hard.

Moreover, Lemma 4.1 implies that it is NP-hard to approximate BSubNBC within a factor less than $4/3$. The subtle point here is that given a subnetwork H, we do not know how to efficiently compute the worst equilibrium bottleneck cost $B(H, r)$. However, we can use the notion of a good subnetwork of D and deal with this issue. Specifically, let A be any approximation algorithm for BSubNBC with approximation ratio less than $4/3$. Then, if D is a YES-instance of 2-DDP, A applied to the network G, constructed in the proof of Lemma 4.1, returns a subnetwork H with $B(H, r) < r/3$. Thus, by Lemma 4.1, H contains a good subnetwork of D, which can be checked in polynomial time. If D is a NO-instance, D contains no good subnetworks. Hence, the outcome of A would allow us to distinguish between YES and NO instances of 2-DDP.

Remark 4.3. If we let the edges to have more general latency functions, such as polynomials of greater degree or exponential functions, then we can get greater inapproximability factors for BSubNBC.

For example, if we use the cost functions x^d instead of x and $x^d/2^d$ instead of $x/2$ in network G in the proof of Lemma 4.1 (fig. 4.1), then we will get an inapproximability ratio of $(4/3)^d$.

Using a^x instead of x and $a^{x/2}$ instead of $x/2$ in the proof of Lemma 4.1 (fig. 4.1), we get an inapproximability ratio of $a^{r/12} = B^*(\mathcal{G})^{1/3}$ (depending on a and r).

4.4 Approximating the Best Subnetwork is Hard

Next, we apply essentially the same construction as in the proof of Lemma 4.1, but in a recursive way, and show that it is NP-hard to approximate BSubNBC for linear bottleneck routing games within a factor of $O(n^{.121-\varepsilon})$, for any constant $\varepsilon > 0$. Throughout this section, we let $\mathcal{I} = (D, s_1, s_2, t_1, t_2)$ be a 2-DDP instance, and let G be an $s-t$ network, which includes (possibly many copies of) D and can be constructed from $\mathcal{I}$ in polynomial time. We assume that G has a linear latency function $d_e(x) = a_e x$, $a_e > 0$, on each edge e, and for any traffic rate $r > 0$, the bottleneck routing game $\mathcal{G} = (G, c, r)$ has $B^*(\mathcal{G}) = r/\gamma_1$, for some $\gamma_1 > 0$. Moreover,

1. If $\mathcal{I}$ is a YES-instance of 2-DDP, there exists a subnetwork H of G with $B(H, r) = r/\gamma_1$.

2. If $\mathcal{I}$ is a NO-instance of 2-DDP, for all subnetworks H' of G, $B(H', r) \geq r/\gamma_2$, for a $\gamma_2 \in (0, \gamma_1)$.

3. For all subnetworks H' of G, either H' contains at least one copy of a good subnetwork of D, or $B(H', r) \geq r/\gamma_2$.

The existence of such a network shows that it is NP-hard to approximate BSubNBC within a factor less than $\gamma = \gamma_1/\gamma_2$. Thus, we usually refer to G as a γ-gap instance (with linear latencies). For example, for the network G in the proof of Lemma 4.1, $\gamma_1 = 4$ and $\gamma_2 = 3$, and thus G is a $4/3$-gap instance. We next show that given $\mathcal{I}$ and a γ_1/γ_2-gap instance G, we can construct a $(4\gamma_1)/(3\gamma_2)$-gap instance G', i.e., we can amplify the inapproximability gap by a factor of $4/3$.

Lemma 4.4. Let $\mathcal{I} = (D, s_1, s_2, t_1, t_2)$ be a 2-DDP instance, and let G be a γ_1/γ_2-gap instance with linear latencies, based on $\mathcal{I}$. Then, we can construct, in time polynomial in the size of $\mathcal{I}$ and G, an $s-t$ network G' with a linear latency function $d_e(x) = a_e x$, $a_e > 0$, on each edge e, so that for any traffic rate $r > 0$, the bottleneck routing game $\mathcal{G}' = (G', c, r)$ has $B^*(\mathcal{G}) = r/(4\gamma_1)$, and:

1. If $\mathcal{I}$ is a YES-instance of 2-DDP, there exists a subnetwork H of G' with $B(H, r) = r/(4\gamma_1)$.

2. If $\mathcal{I}$ is a NO-instance of 2-DDP, for every subnetwork H' of G', $B(H', r) \geq r/(3\gamma_2)$.

3. For all subnetworks H' of G', either H' contains at least one copy of a good subnetwork of D, or $B(H', r) \geq r/(3\gamma_2)$.

Proof. Starting from D, we obtain G' by applying the construction of Lemma 4.1, but with all external edges, except for e_2, replaced by a copy of the gap-instance G. For convenience, we refer to the copy of the gap-instance replacing the external edge e_i, $i \in \{1, 3, \ldots, 9\}$, as the *edgework* G_i. Formally, to obtain G', we start from D and add four new vertices, s, t, v, u. We connect s to u, with the $s-u$ edgework G_1, and v to t, with the $s-u$ edgework G_3, where in both G_1 and G_3, we replace the latency function $d_e(x)$ of each edge e in the gap instance with $d_e(x)/2$ (this is because in Lemma 4.1, the external edges e_1 and e_3 have latencies $x/2$). Moreover, instead of the external edge e_i, $i \in \{4, \ldots, 9\}$, we connect (s, v), (v, s_1), (s, s_2), (t_1, u), (u, t), and (t_2, t) with the edgework G_i. The latencies in these edgeworks are as in the gap instance. Furthermore, we add the external edge $e_2 = (u, v)$ with latency $d_{e_2}(x) = \varepsilon x$, for some $\varepsilon \in (0, \frac{1}{4\gamma_1})$ (see also Fig. 4.3.a). Also, each edge e of D has latency $d_e(x) = \varepsilon x$. We next consider the corresponding routing instance $\mathcal{G}'$ with an arbitrary traffic rate $r > 0$. Throughout the proof, when we define a routing instance, we omit, for simplicity, the coordinate c, referring to the latency functions, with the understanding that they are defined as above.

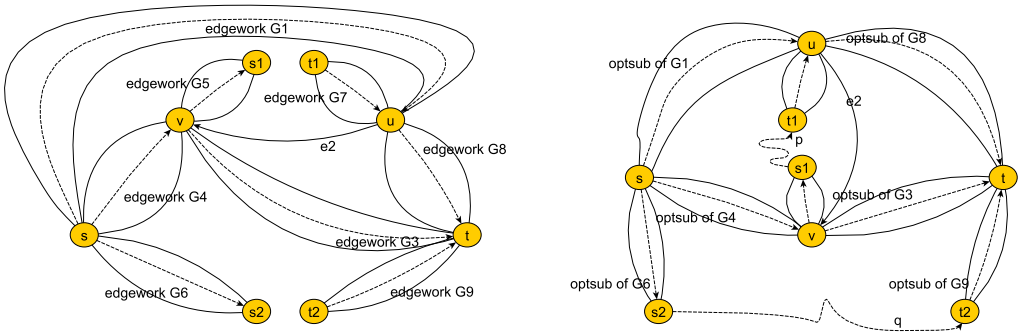

Figure 4.3: (a) The network G' constructed in the proof of Lemma 4.4. The structure of G' is similar to the structure of the network G in Fig. 4.1, with each external edge e_i, except for e_2, replaced by the edgework G_i. (b) The structure of a best subnetwork H of G', with PoA = 1, when D contains a pair of vertex-disjoint paths, p and q, connecting s_1 to t_1 and s_2 to t_2. To complete H, we use an optimal subnetwork (or simply, subedgework) of each edgework G_i.

Intuitively, each G_i, $i \in \{4, \ldots, 9\}$, behaves as an external edge (hence the term edge(net)work), which at optimality has a bottleneck cost of r/γ_1, for any traffic rate r entering G_i. Moreover, if $\mathcal{I}$ is a YES-instance of 2-DDP, the edgework G_i has a subedgework H_i for which $B(H_i, r) = r/\gamma_1$, for any r, while if H_i does not contain any copies of a good subnetwork of D (or, if $\mathcal{I}$ is a NO-instance), for all subedgeworks H_i' of G_i, $B(H_i', r) \geq r/\gamma_2$, for any r.

The same holds for G_1 and G_3, but with a worst equilibrium bottleneck cost of $r/(2\gamma_1)$ in the former case, and of $r/(2\gamma_2)$ in the latter case, because the latency functions of G_1 and G_3 are scaled by $1/2$ (see also Proposition 1).

The proofs of the following propositions are conceptually similar to the proofs of the corresponding claims in the proof Lemma 4.1.

Proposition 4. *The optimal bottleneck cost of $\mathcal{G}'$ is $B^*(\mathcal{G}') = r/(4\gamma_1)$.*

Proof. We have to show that $B^*(\mathcal{G}') = r/(4\gamma_1)$. For the upper bound, as in the proof of Lemma 4.1, we assume that D contains an $s_1 - t_2$ path p and an $s_2 - t_1$ path q, which are edge-disjoint. We route (i) $r/4$ units of flow through the edgeworks G_4, G_5, next through the path p, and next through the edgework G_9, (ii) $r/4$ units through the edgeworks G_6, next through the path q, and next through the edgeworks G_7 and G_8, and (ii) $r/2$ units through the edgework G_1, next through the external edge e_2, and next through the edgework G_3. These routes are edge(work)-disjoint, and if we route the flow optimally through each edgework, the bottleneck cost is $r/(4\gamma_1)$. As for the lower bound, we observe that the edgeworks H_1, H_4, and H_6 essentially form an $s-t$ cut in G', and thus every feasible flow has a bottleneck cost of at least $r/(4\gamma_1)$. □

Proposition 5. *If $\mathcal{I}$ is a* YES*-instance, there is a subnetwork H of G' with $B(H, r) = r/(4\gamma_1)$.*

Proof. If $\mathcal{I}$ is a YES-instance of 2-DDP, then (i) there are two vertex-disjoint paths in D, p and q, connecting s_1 to t_1 and s_2 to t_2, and (ii) there is an optimal subnetwork (or simply, subedgework) H_i of each edgework G_i so that for any traffic rate r routed through H_i, the worst equilibrium bottleneck cost $B(H_i, r)$ is r/γ_1, if $i \in \{4, \ldots, 9\}$, and $r/(2\gamma_1)$, if $i \in \{1, 3\}$. Let H be the subnetwork of G' that consists of only the edges of the paths p and q from D, of the external edge e_2, and of the optimal subedgeworks H_i, $i \in \{1, 3, \ldots, 9\}$ (see also Fig. 4.3.b). We observe that we can route: (i) $r/4$ units of flow through the subedgeworks H_4, H_5, next through the path p, and next through the subedgeworks H_7 and H_8, (ii) $r/4$ units of flow through the subedgework H_6, next through the path q, and next through the subedgework H_9, and (iii) $r/2$ units of flow through the subedgework H_1, next through the external edge e_2, and next through the subedgework H_3. These routes are edge(work)-disjoint, and if we use any Nash flow through each of the routing instances $(H_i, r/4)$, $i \in \{4, \ldots, 9\}$, $(H_1, r/2)$, and $(H_3, r/2)$, we obtain a Nash flow of the instance (H, r) with a bottleneck cost of $r/(4\gamma_1)$.

We next show that any Nash flow of (H, r) has a bottleneck cost of at most $r/(4\gamma_1)$. To reach a contradiction, let us assume that some feasible

Nash flow f has bottleneck cost $B(f) > r/(4\gamma_1)$. We recall that f is a Nash flow iff the edges of G' with bottleneck cost $B(f) > r/(4\gamma_1)$ form an $s-t$ cut. This cut does not include the edges of the paths p and q and the external edge e_2, due to the choice of their latencies. Hence, this cut includes a similar cut either in H_6 or in H_9 (or in both), either in H_1 or H_3 (or in both), and either in H_4 or in H_8 (or in H_5 or in H_6, in certain combinations with other subedgeworks, see also Fig. 4.3.b). Let us consider the case where the edges with bottleneck cost $B(f) > r/(4\gamma_1)$ form a cut in H_1, H_4, and H_6. Namely, the edges of H_1, H_4, and H_6, with bottleneck cost equal to $B(f) > r/(4\gamma_1)$ form an $s-u$, an $s-v$, and an $s-s_2$ cut, respectively, and thus the restriction of f to each of H_1, H_4, and H_6, is an equilibrium flow of bottleneck cost greater than $r/(4\gamma_1)$ for the corresponding routing instance. Since $\mathcal{I}$ is a YES-instance, this can happen only if the flow through H_1 is more than $r/2$, and the flow through each of H_4 and H_6 is more than $r/4$ (see also property (ii) of optimal subedgeworks above). Hence, we obtain that more than r units of flow leave s, a contradiction. All other cases are similar. □

The most technical part of the proof is to show (3), namely that for any subnetwork H' of G', if H' does not contain any copies of a good subnetwork of D, then $B(H', r) \geq r/(3\gamma_2)$. This immediately implies (2), since if $\mathcal{I}$ is a NO-instance of 2-DDP, D includes no good subnetworks. To prove (3), we consider any subnetwork H' of G', and let H'_i be the subedgework of each G_i present in H'. We assume that the subedgeworks H'_i do not contain any copies of a good subnetwork of D, and show that if the subnetwork of D connecting s_1 and s_2 to t_1 and t_2 in H' is also bad, then $B(H', r) \geq r/(3\gamma_2)$.

At the technical level, we repeatedly use the idea of a flow f_i through a subedgework H'_i that "saturates" H'_i, in the sense that f_i is a Nash flow with bottleneck cost at least $r_i/(3\gamma_2)$ for the subinstance (H'_i, r_i). Formally, we say that a flow rate r_i *saturates* a subedgework H'_i if $B(H'_i, r_i) \geq r_i/(3\gamma_2)$. We refer to the flow rate r_i^s for which $B(H'_i, r_i^s) = r_i^s/(3\gamma_2)$ as the *saturation rate* of H'_i. We note that the saturation rate r_i^s is well-defined, because the latency functions of G_is are linear and strictly increasing. Moreover, by property (3) of gap instances, the saturation rate of each subedgework H'_i is $r_i^s \leq r/3$, if $i \in \{4, \ldots, 9\}$, and $r_i^s \leq 2r/3$, if $i \in \{1, 3\}$. Thus, at the intuitive level, the subedgeworks H'_i behave as the external edges of the network constructed in the proof of Lemma 4.1. Hence, to show that $B(H', r) \geq r/(3\gamma_2)$, we need to construct a flow of rate (at most) r that saturates a collection of subedgeworks comprising an $s-t$ cut in H'.

Our first step in this direction is to simplify the possible structure of H'.

Proposition 6. *Let H' be any subnetwork of G' whose subedgeworks H'_i do not contain any copies of a good subnetwork of D. Then, either the subnetwork H' contains (i) the external edge e_2, (ii) at least one path outgoing from each of s_1 and s_2 to either t_1 or t_2, and (iii) at least one path incoming to each of t_1 and t_2 from either s_1 or s_2, or $B(H', r) \geq r/(3\gamma_2)$.*

Proof. For convenience, in the proofs of Proposition 6 and Proposition 7, we slightly abuse the terminology, and say that a collection of subedgeworks of H' form an $s-t$ cut, if the union of any cuts in them comprises an $s-t$ cut in H'. Moreover, whenever we write that r_i units of flow are routed through a subedgework H_i, we assume that the routing through H_i corresponds to the worst Nash flow of (H_i, r_i). Also, we recall that since subedgeworks H'_i do not contain any copies of a good subnetwork of D, by property (3) of gap instances, the saturation rate of each H'_i is $r^s_i \leq r/3$, if $i \in \{4, \ldots, 9\}$, and $r^s_i \leq 2r/3$, if $i \in \{1, 3\}$.

We start by showing that either the external edge e_2 is present in H', or $B(H', r) \geq r/(3\gamma_2)$. Indeed, if e_2 is not present in H', the subedgeworks H'_4, H'_6, and H'_8 form an $s-t$ cut in H'. Therefore, we can construct a Nash flow f that routes at least $r/3$ units of flow through H'_4, H'_6, and H'_8, and has $B(f) \geq r/(3\gamma_2)$. Therefore, we can assume, without loss of generality, that e_2 is present in H'.

Similarly, we show that either H' includes at least one path outgoing from s_2 to either t_1 or t_2, and at least one path incoming to t_2 from either s_1 or s_2, or $B(H', r) \geq r/(3\gamma_2)$. In particular, if s_2 is connected to neither t_1 nor t_2, the subedgeworks H'_1 and H'_4 form an $s-t$ cut in H'. Thus, we can construct a Nash flow f that saturates the subedgework H'_1 (or the subedgeworks H'_3 and H'_8, if $r^s_1 > r^s_3 + r^s_8$) and the subedgework H'_4 (or the subedgeworks H'_3 and either H'_5, or H'_9 and at least one of the H'_7 and H'_8, depending on r^s_4 and the saturation rates of the rest). We note that this is always possible with r units of flow, because $r^s_1 \leq 2r/3$ and $r^s_4 \leq r/3$. Therefore, the bottleneck cost of f is $B(f) \geq r/(3\gamma_2)$. In case where there is no path incoming to t_2 from either s_1 or s_2, the subedgeworks H'_3 and H'_8 form an $s-t$ cut in H'. As before, we can construct a Nash flow f that saturates the subedgeworks H'_3 and H'_8 (or, as before, an appropriate combination of other subedgeworks carrying flow to H'_3 and H'_8), and has $B(f) \geq r/(3\gamma_2)$. Therefore, we can assume, without loss of generality, that H' includes at least one path outgoing from s_2 to either t_1 or t_2, and at least one path incoming to t_2 from either s_1 or s_2.

Next, we show that either H' includes at least one path outgoing from s_1 to either t_1 or t_2, and at least one path incoming to t_1 from either s_1 or s_2, or $B(H', r) \geq r/(3\gamma_2)$. In particular, let us consider the case where s_1

is connected to neither t_1 nor t_2 (see also Fig. 4.4.a, the case where there is no path incoming to t_1 from either s_1 or s_2 can be handled similarly). In the following, we assume that s_2 is connected to t_2 (because, by the analysis above, we can assume that there is a path incoming to t_2, and s_1 is not connected to T_2), and construct a Nash flow f of bottleneck cost $B(f) \geq r/(3\gamma_2)$.

We first route $\min\{r_6^s, r_9^s\} \leq r/3$ units of flow through the subedgework H'_6, next through an $s_2 - t_2$ path, and finally through the subedgework H'_9, and saturate either H'_6 or H'_9 (or both). If there is an $s_2 - t_1$ path and H'_6 is not saturated, we keep routing flow through H'_6, next through an $s_2 - t_1$ path, and next through the subedgeworks H'_7 and H'_8, until either the subedgework H'_6 or at least one of the subedgeworks H'_7 and H'_8 become saturated. Thus, we saturate at least one edgework on every $s-t$ path that includes s_2.

Next, we show how to saturate at least one edgework on every $s-t$ path that includes either v or u. If $r_1^s \leq r_3^s \leq 2r/3$, we route r_1^s units of flow through H'_1, e_2, and H'_3, and route $\min\{r_3^s - r_1^s, r_4^s\}$ units of flow through H'_4 and H'_3, and saturate either H'_1 and H'_3 or H'_1 and H'_4. If $r_3^s < r_1^s \leq 2r/3$, we route r_3^s units of flow through H'_1, e_2, and H'_3, and route $\min\{r_3^s - r_1^s, r_8^s\}$ units of flow through H'_1 and H'_8, and saturate either H'_1 and H'_3 or H'_3 and H'_8.

The remaining flow (if any) can be routed through these routes, in proportional rates. In all cases, we obtain an $s-t$ cut consisting of saturated subedgeworks. Thus, the resulting flow f is a Nash flow with a bottleneck cost of at least $r/(3\gamma_2)$.

□

Now, let us focus on a subnetwork H' of G' that contains (i) the external edge e_2, (ii) at least one path outgoing from each of s_1 and s_2 to either t_1 or t_2, and (iii) at least one path incoming to each of t_1 and t_2 from either s_1 or s_2. If the copy of the subnetwork of D connecting s_1 and s_2 to t_1 and t_2 in H' is also bad, properties (ii) and (iii) imply that H' contains an $s_1 - t_2$ path p and an $s_2 - t_1$ path q. In this case, the entire subnetwork H' essentially behaves as if it included all edges of G'. Then, a routing similar to that in Fig. 4.2.c gives a Nash flow with a bottleneck cost of $r/(3\gamma_2)$. This intuition is formalized by the following proposition.

Proposition 7. *Let H' be any subnetwork of G' that satisfies (i), (ii), and (iii) above, and does not contain any copies of a good subnetwork of D. Then $B(H', r) \geq r/(3\gamma_2)$.*

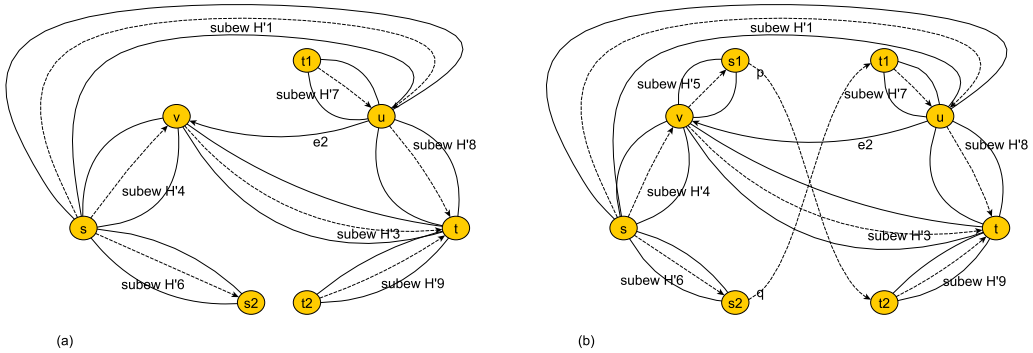

Figure 4.4: The structure of possible subnetworks of G' when there is no pair of vertex-disjoint paths connecting s_1 to t_1 and s_2 to t_2. The subnetwork (a) contains a path outgoing from s_2 to either t_1 or t_2, and no path outgoing from s_1 to either t_1 or t_2. Hence, no flow can be routed through the edgework G_5, and thus we can regard G_5 as being absent from H'. The subnetwork (b) essentially corresponds to the case where all edges of G' are present in H'.

Proof. In the following, we consider a subnetwork H' of G' which does not include any copies of a good subnetwork of D, and contains (i) the external edge e_2, (ii) at least one path outgoing from each of s_1 and s_2 to either t_1 or t_2, and (iii) at least one path incoming to each of t_1 and t_2 from either s_1 or s_2. Since the copy of the subnetwork of D connecting s_1 and s_2 to t_1 and t_2 in H' is bad, properties (ii) and (iii) imply that H' contains an $s_1 - t_2$ path p and an $s_2 - t_1$ path q. Moreover, since the subedgeworks H'_i do not include any copies of a good subnetwork of D, by property (3) of gap instances, the saturation rate of each H'_i is $r^s_i \leq r/3$, if $i \in \{4, \ldots, 9\}$, and $r^s_i \leq 2r/3$, if $i \in \{1, 3\}$.

We next show that for such a subnetwork H', we can construct a Nash flow f of bottleneck cost $B(f) \geq r/(3\gamma_2)$. At the conceptual level, as in the last case in the proof of Lemma 4.1, we seek to construct a Nash flow by routing $r/3$ units of flow through each of the following three routes: (i) H'_1, e_2, and H'_3, (ii) H'_1, e_2, H'_5, p, and H'_9, and (iii) H'_6, q, H'_7, e_2, and H'_3. However, for simplicity of the analysis, we regard the corresponding (edge) flow as being routed through just two routes: a rate of $2r/3$ is routed through H'_1, e_2, and H'_3, and a rate of $r/3$ is routed through the (possibly non-simple) route H'_6, q, H'_7, e_2, H'_5, p, and H'_9. We do so because the latter routing allows us to consider fewer cases in the analysis. We conclude the proof by showing that if the latter route is not simple, we can always decompose the flow into the three simple routes above.

In the following, we assume that with a flow rate of at most $2r/3$, routed through H'_1, e_2, and H'_3 (and possibly through H'_4 and H'_8), we can saturate

both subedgeworks H'_1 and H'_3. Otherwise, as in the last case in the proof of Proposition 6, we can show how with a total flow rate of at most $2r/3$, part of which is routed through either H'_4 or H'_8, we can saturate either H'_1 and H'_4, or H'_3 and H'_8. Then, the remaining $r/3$ units of flow can saturate either H'_6, in the former case, or H'_9, in the latter case. Thus, we obtain a Nash flow with a bottleneck cost of at least $r/(3\gamma_2)$.

Having saturated both subedgeworks H'_1 and H'_3, using at most $2r/3$ units of flow, we have at least $r/3$ units of flow to saturate the subedgeworks H'_5, H'_6, H'_7, and H'_9, or an appropriate subset of them, so that together with H'_1 and H'_3, they form an $s-t$ cut in H'. We first route $\tau \equiv \min\{r^s_5, r^s_6, r^s_7, r^s_9\} \leq r/3$ units of flow through H'_6, q, H'_7, e_2, H'_5, p, and H'_9, until t, and consider different cases, depending on which of the subedgeworks H'_5, H'_6, H'_7, and H'_9 has the minimum saturation rate.

- If $\tau = r^s_9$, H'_9 is saturated. We first assume that H' contains an $s_1 - t_1$ path, and route (some of) the remaining flow (i) through H'_4, H'_5, an $s_1 - t_1$ path, H'_7, and H'_8, and (ii) through H'_6, q, H'_7, and H'_8. We do so until either at least one of the subedgeworks H'_7 and H'_8 or the subedgework H'_6 and at least one of the subedgeworks H'_4 and H'_5 become saturated. Since $\min\{r^s_7, r^s_8\} \leq r/3$, this requires at most $r/3 - \tau$ additional units of flow. If H' does not contain an $s_1 - t_1$ path, we route the remaining flow only through route (ii), until either at least one of the subedgeworks H'_7 and H'_8 or the subedgework H'_6 become saturated. In both cases, the newly saturated subedgeworks, together with the saturated subedgeworks H'_1, H'_3, and H'_9, form an $s-t$ cut of saturated subedgeworks, and thus the worst equilibrium bottleneck cost is at least $r/(3\gamma_2)$.

- If $\tau = r^s_6$, H'_6 is saturated. As before, we first assume that H' contains an $s_1 - t_1$ path, and route the remaining flow (i) through H'_4, H'_5, p, and H'_9, and (ii) through H'_4, H'_5, an $s_1 - t_1$ path, H'_9 and H'_8, until either at least one of the subedgeworks H'_4 and H'_5, or the subedgework H'_9 and at least one of the subedgeworks H'_7 and H'_8 become saturated. Since $\min\{r^s_4, r^s_5\} \leq r/3$, this requires at most $r/3 - \tau$ additional units of flow. If H' does not contain an $s_1 - t_1$ path, we route the remaining flow only through route (i), until either at least one of the subedgeworks H'_4 and H'_5 or the subedgework H'_9 become saturated. In both cases, the newly saturated subedgeworks, together with the saturated subedgeworks H'_1, H'_3, and H'_6, form an $s-t$ cut of saturated subedgeworks, and thus the worst equilibrium bottleneck cost is at least $r/(3\gamma_2)$.

- If $\tau = r^s_7$, H'_7 is saturated. Then, we first assume that H' contains

an $s_2 - t_2$ path, and route the remaining flow (i) through H'_4, H'_5, p, and H'_9, and (ii) through H'_6, an $s_2 - t_2$ path, and H'_9, until either the subedgework H'_9, or the subedgework H'_6 and at least one of the subedgeworks H'_4 and H'_5 become saturated. Since $r^s_9 \leq r/3$, this requires at most $r/3 - \tau$ additional units of flow. If H' does not contain an $s_2 - t_2$ path, we route the remaining flow only through route (i), until either at least one of the subedgeworks H'_4 and H'_5 or the subedgework H'_9 become saturated. In both cases, the newly saturated subedgeworks, together with the saturated subedgeworks H'_1, H'_3, and H'_7, form an $s - t$ cut of saturated subedgeworks, and thus the worst equilibrium bottleneck cost is at least $r/(3\gamma_2)$.

- If $\tau = r^s_5$, H'_5 is saturated. As before, we first assume that H' contains an $s_2 - t_2$ path, and route the remaining flow (i) through H'_6, q, H'_7, and H'_8, and (ii) through H'_6, an $s_2 - t_2$ path, and H'_9, until either the subedgework H'_6, or the subedgework H'_9 and at least one of the subedgeworks H'_7 and H'_8 become saturated. Since $r^s_6 \leq r/3$, this requires at most $r/3 - \tau$ additional units of flow. If H' does not contain an $s_2 - t_2$ path, we route the remaining flow only through route (i), until either at least one of the subedgeworks H'_7 and H'_8 or the subedgework H'_6 become saturated. In both cases, the newly saturated subedgeworks, together with the saturated subedgeworks H'_1, H'_3, and H'_5, form an $s - t$ cut of saturated subedgeworks, and thus the worst equilibrium bottleneck cost is at least $r/(3\gamma_2)$.

Thus, in all cases, we obtain an equilibrium flow with a bottleneck cost of at least $r/(3\gamma_2)$. However, in the construction above, the route H'_6, q, H'_7, e_2, H'_5, p, H'_9 may not be simple, since p and q may not be vertex-disjoint. If this is the case, this route is technically not allowed by our model, where the flow is only routed through simple $s - t$ paths. Nevertheless, the corresponding edge flow can be decomposed into the following three simple routes: (i) H'_1, e_2, and H'_3, (ii) H'_1, e_2, H'_5, p, and H'_9, and (iii) H'_6, q, H'_7, e_2, and H'_3, unless $\min\{r^s_1, r^s_3\} \leq r/3$. Moreover, if $\min\{r^s_1, r^s_3\} \leq r/3$, we can work as above, and saturate both H'_1 and H'_3 with at most $r/3$ units of flow. The remaining $2r/3$ units of flow can be routed (i) through H'_6, q, H'_7, and H'_8, and (ii) through H'_4, H'_5, p, and H'_9, and possibly either through H'_6, an $s_2 - t_2$ path[1], and H'_9, or through H'_4, H'_5, an $s_1 - t_1$ path, H'_7, and H'_8, until either H'_4 (or H'_5) and H'_6, or H'_7 (or H'_8) and H'_9 are saturated. This routing only uses simple routes. In addition,

[1]We note that if the paths p and q are not vertex-disjoint, we also have an $s_1 - t_1$ path and an $s_2 - t_2$ path in H'.

these saturated subedgeworks, together with the saturated subedgeworks H_1' and H_3', form an $s-t$ cut of saturated subedgeworks, and thus the worst equilibrium bottleneck cost is at least $r/(3\gamma_2)$. □

Propositions 6 and 7 immediately imply part (3) of the lemma, which, in turn, implies part (2). □

Each time we apply Lemma 4.4 to a γ-gap instance G, we obtain a $4\gamma/3$-gap instance G' with a number of vertices of at most 8 times the vertices of G plus the number of vertices of D. Therefore, if we start with an instance $\mathcal{I} = (D, s_1, s_2, t_1, t_2)$ of 2-DDP, where D has k vertices, and apply Lemma 4.1 once, and subsequently apply Lemma 4.4 for $\lfloor \log_{4/3} k \rfloor$ times, we obtain a k-gap instance $\mathcal{G}'$, where the network G' has $n = O(k^{8.23})$ vertices. Suppose now that there is a polynomial-time algorithm A that approximates the best subnetwork of G' within a factor of $O(k^{1-\varepsilon}) = O(n^{0.121-\varepsilon})$, for some small $\varepsilon > 0$. Then, if $\mathcal{I}$ is a YES-instance of 2-DDP, algorithm A, applied to G', should return a best subnetwork H with at least one copy of a good subnetwork of D. Since H contains a polynomial number of copies of subnetworks of D, and we can check whether a subnetwork of D is good in polynomial time, we can efficiently recognize $\mathcal{I}$ as a YES-instance of 2-DDP. On the other hand, if $\mathcal{I}$ is a NO-instance of 2-DDP, D includes no good subnetworks. Again, we can efficiently check that in the subnetwork returned by algorithm A, there are not any copies of a good subnetwork of D, and hence recognize $\mathcal{I}$ as a NO-instance of 2-DDP. Thus, we obtain that:

Theorem 4.5. For bottleneck routing games with strictly increasing linear latencies, it is NP-hard to approximate BSubNBC within a factor of $O(n^{0.121-\varepsilon})$, for any constant $\varepsilon > 0$.

Remark 4.6. If in the network G in the proof of Lemma 4.1 (fig. 4.1) we replace the cost functions x and $x/2$ with x^d and $x^d/2^d$ respectively, we will get an instance with $\gamma_1 = 4^d$ and $\gamma_2 = 3^d$, and thus G would be a $(4/3)^d$-gap instance. Moreover, if we apply the same techniques as in lemma 4.4, we can amplify the inaproximability gap. As in Lemma 4.4 we inductively create a new network using as a base the base network of figure 4.1 with cost functions x^d instead of x and $x^d/2^d$ instead of $x/2$. In the new network the edges $e_4, \ldots, e_9$ are replaced with a copy of the old network with the known gap $(4/3)^d$. Edges e_1, e_3 are replaced with a copy of the old network but with all the cost functions divided by 2^d. This will result to a graph that gives an inapproximability gap of $(4/3)^{2d}$. Doing this $t = \log_{(4/3)^d} n$ times, we result to a network with $O(n^{\log_{(4/3)^d} 8+1})$ vertices and an inapproximability gap of n. So, in a similar way like before, we get

that (unless P=NP), we cannot polynomially approximate BSubNBC within a factor of $O(n^{1/(\log_{(4/3)^d} 8+1)} - \varepsilon)$

Using a^x instead of x and $a^{x/2}$ instead of $x/2$ and applying the same technique for say t times we get a network with gap $B^*(\mathcal{G})^{(4/3)^t-1}$

4.5 Networks with Quasipolynomially Many Paths

In this section, we approximate, in quasipolynomial-time, the best subnetwork and its worst equilibrium bottleneck cost for instances $\mathcal{G} = (G, c, r)$ where the network G has quasipolynomially many $s-t$ paths, the latency functions are continuous and satisfy a Lipschitz condition, and the worst Nash flow in the best subnetwork routes a non-negligible amount of flow on all used edges.

We highlight that the restriction to networks with quasipolynomially many $s-t$ paths is somehow necessary, in the sense that Theorem 4.5 shows that if the network has exponentially many $s-t$ paths, as it happens for the hard instances of 2-DDP, and thus for the networks G and G' constructed in the proofs of Lemma 4.1 and Lemma 4.4, it is NP-hard to approximate BSubNBC within any reasonable factor. Also, we can always assume, without loss of generality, that the worst Nash flow of the best subnetwork H^* assigns positive flow to all edges of H^*. Otherwise, we can remove any unused edges, without increasing the worst equilibrium bottleneck cost of H^*. In addition, we assume here that there is a constant $\delta > 0$, such that the worst Nash flow in H^* routes more than δ units of flow on all edges of the best subnetwork H^*.

In the following, we normalize the traffic rate r to 1. This is for convenience and can be made without loss of generality[2]. Our algorithm is based on [42, Lemma 2], which applies Althöfer's "Sparsification" Lemma [5] (similar technique in [60], independently), and shows that any flow can be approximated by a "sparse" flow using logarithmically many paths.

Lemma 4.7. Let $\mathcal{G} = (G(V, E), c, 1)$ be a routing instance, and let f be any $\mathcal{G}$-feasible flow. Then, for any $\varepsilon > 0$, there exists a $\mathcal{G}$-feasible flow $\tilde{f}$ using at most $k(\varepsilon) = \lfloor \log(2m)/(2\varepsilon^2) \rfloor + 1$ paths, such that for all edges e, $|\tilde{f}_e - f_e| \leq \varepsilon$, if $f_e > 0$, and $\tilde{f}_e = 0$, otherwise.

By Lemma 4.7, there exists a sparse flow $\tilde{f}$ that approximates the worst Nash flow f on the best subnetwork H^* of G. Moreover, the proof of [42,

[2] Given a bottleneck routing game $\mathcal{G}$ with traffic rate $r > 0$, we can replace each latency function $d_e(x)$ with $d_e(rx)$, and obtain a bottleneck routing game $\mathcal{G}'$ with traffic rate 1, and the same Nash flows, PoA, and solutions to BSubNBC.

Lemma 2] shows that the flow $\tilde{f}$ is determined by a multiset P of at most $k(\varepsilon)$ paths, selected among the paths used by f. Then, for every path $p \in \mathcal{P}$, $\tilde{f}_p = |P(p)|/|P|$, where $|P(p)|$ is number of times the path p is included in the multiset P, and $|P|$ is the cardinality of P. Therefore, if the total number $|\mathcal{P}|$ of $s-t$ paths in G is quasipolynomial, we can find, in quasipolynomial-time, by exhaustive search, a flow-subnetwork pair that approximates the optimal solution of BSubNBC. Based on this intuition, we next obtain an approximation algorithm for BSubNBC on networks with quasipolynomially many paths, under the assumption that there is a constant $\delta > 0$, such that the worst Nash flow in the best subnetwork H^* routes more than δ units of flow on all edges of H^*. This assumption is necessary so that the exhaustive search on the family of sparse flows of Lemma 4.7 can generate the best subnetwork H^*, which is crucial for the analysis.

Theorem 4.8. Let $\mathcal{G} = (G(V,E), c, 1)$ be a bottleneck routing game with continuous latency functions that satisfy the Lipschitz condition with a constant $\xi > 0$, let H^* be the best subnetwork of G, and let f^* be the worst Nash flow in H^*. If for all edges e of H^*, $f^*_e > \delta$, for some constant $\delta > 0$, then for any constant $\varepsilon > 0$, we can compute in time $|\mathcal{P}|^{O(\log(2m)/\min\{\delta^2, \varepsilon^2/\xi^2\})}$ a flow f and a subnetwork H such that: (i) f is an $\varepsilon/2$-Nash flow in the subnetwork H, (ii) $B(f) \leq B(H^*, 1) + \varepsilon$, (iii) $B(H, 1) \leq B(f) + \varepsilon/4$, and (iv) $B(f) \leq B(H, 1) + \varepsilon/2$.

Proof. Let $\varepsilon > 0$ be a constant, and let $\epsilon_1 = \min\{\delta, \varepsilon/(4\xi)\}$, and $\epsilon_2 = \varepsilon/2$. We show that a flow-subnetwork pair (H, f) with the desired properties can be computed in time $|\mathcal{P}|^{O(k(\epsilon_1))}$, where $k(\varepsilon_1) = \lfloor \log(2m)/\min\{2\delta^2, \varepsilon^2/(8\xi^2)\} \rfloor + 1$, For convenience, we say that a flow g is a *candidate flow* if there is a multiset P of paths from $\mathcal{P}$, with $|P| \leq k(\epsilon_1)$, such that $g_p = |P(p)|/|P|$, for each $p \in \mathcal{P}$. Namely, a candidate flow belongs to the family of sparse flows, which by Lemma 4.7, can approximate any other flow. Similarly, a subnetwork H is a *candidate subnetwork* if there is a candidate flow g such that H consists of the edges used by g (and only of them), and a subnetwork-flow pair (H, g) is a *candidate solution*, if g is a candidate flow, H is a candidate subnetwork that includes all the edges used by g (and possibly some other edges), and g is an ϵ_2-Nash flow in H.

By exhaustive search, in time $|\mathcal{P}|^{O(k(\epsilon_1))}$, we generate all candidate flows, all candidate subnetworks, and compute the bottleneck cost $B(g)$ of any candidate flow g. Then, for each pair (H, g), where g is a candidate flow and H is a candidate subnetwork, we check, in polynomial time, whether g is an ϵ_2-Nash flow in H, and thus whether (H, g) is a candidate solution. Thus, in time $|\mathcal{P}|^{O(k(\epsilon_1))}$, we determine all candidate solutions. For each candidate subnetwork H that participates in at least one candidate solution, we let

$\tilde{B}(H)$ be the maximum bottleneck cost $B(g)$ of a candidate flow g for which (H, g) is a candidate solution. The algorithm returns the subnetwork H that minimizes $\tilde{B}(H)$, and a flow f for which (H, f) is a candidate solution and $\tilde{B}(H) = B(f)$.

The exhaustive search above can be implemented in $|\mathcal{P}|^{O(k(\epsilon_1))}$ time. As for the properties of the solution (H, f), the definition of candidate solutions immediately implies (i), i.e., that f is an $\varepsilon/2$-Nash flow in H.

In the following we use Lemma 4.7, and show (ii), (iii), and (iv).

We first show (ii), i.e., that $B(f) \leq B(H^*, 1) + \varepsilon$. We recall that H^* denotes the best subnetwork of $\mathcal{G}$ and f^* denotes the worst Nash flow in H^*. Also, by hypothesis, $f^*_e > \delta > 0$, for all edges e of H^*.

By Lemma 4.7, there is a candidate flow $\tilde{f}$ such that for all edges e of H^*, $|\tilde{f}_e - f^*_e| \leq \epsilon_1$. Thus, since $\epsilon_1 \leq \delta$, H^* is a candidate network, because $\tilde{f}_e > 0$ for all edges e of H^*. Moreover, by the Lipschitz condition and the choice of ϵ_1, for all edges e of H^*, $|d_e(\tilde{f}_e) - d_e(f^*_e)| \leq \varepsilon/4$. Therefore, since f^* is a Nash flow in H^*, $\tilde{f}$ is an ϵ_2-Nash flow in H^*, and thus $(H, \tilde{f})$ is a candidate solution. Furthermore, $|B(\tilde{f}) - B(f^*)| \leq \varepsilon/4$, i.e., the bottleneck cost of $\tilde{f}$ is within an additive term of $\varepsilon/4$ from the worst equilibrium bottleneck cost of H^*. In particular, $B(\tilde{f}) \leq B(H^*, 1) + \varepsilon/4$.

We also need to show that for any other candidate flow g for which (H^*, g) is a candidate solution, $B(g) \leq B(\tilde{f}) + 3\varepsilon/4$, and thus $\tilde{B}(H^*) \leq B(\tilde{f}) + 3\varepsilon/4 \leq B(H^*, 1) + \varepsilon$. To reach a contradiction, let us assume that there is a candidate flow g that is an ϵ_2-Nash flow in H^* and has $B(g) > B(\tilde{f}) + 3\varepsilon/4$. But then, we should expect that there is a Nash flow g' in H^* that closely approximates g and has a bottleneck cost of $B(g') \approx B(g) > B(f^*)$, a contradiction. Formally, since g is an ϵ_2-Nash flow in H^*, the set of edges with $d_e(g_e) \geq B(g) - \varepsilon/2$ comprises an $s - t$ cut in H^*. Then, by the continuity of the latency functions, we can fix a part of the flow routed essentially as in g, so that there is an $s - t$ cut consisting of used edges with latency $B(g) - \varepsilon/2$, and possibly unused edges with latency at least $B(g) - \varepsilon/2$, and reroute the remaining flow on top of it, so that we obtain a Nash flow g' in H^*. But then,

$$B(g') \geq B(g) - \varepsilon/2 > B(\tilde{f}) + \varepsilon/4 \geq B(f^*)\,,$$

which contradicts the hypothesis that f^* is the worst Nash flow in H^*.

Therefore, $\tilde{B}(H^*) \leq B(H^*, 1) + \varepsilon$. Since the algorithm returns the candidate solution (H, f), and not a candidate solution including H^*, $\tilde{B}(H) \leq B(H^*)$. Thus, we obtain (ii), namely that $\tilde{B}(H) = B(f) \leq B(H^*, 1) + \varepsilon$.

We proceed to show (iii), namely that $B(H, 1) \leq B(f) + \varepsilon/4$. To this end, we let g be the worst Nash flow in H. By Lemma 4.7, there is a candidate

flow $\tilde{g}$ such that for all edges e of H, $|\tilde{g}_e - g_e| \leq \epsilon_1$, if $g_e > 0$, and $\tilde{g}_e = 0$, otherwise. Therefore, by the Lipschitz condition and the choice of ϵ_1, for all edges e of H, $|d_e(\tilde{g}_e) - d_e(g_e)| \leq \varepsilon/4$, if $g_e > 0$, and $d_e(\tilde{g}_e) = d_e(g_e) = 0$, otherwise. This implies that $|B(\tilde{g}) - B(g)| \leq \varepsilon/4$, i.e., that bottleneck cost of $\tilde{g}$ is within an additive term of $\varepsilon/4$ from the bottleneck cost of g. In particular, $B(g) \leq B(\tilde{g}) + \varepsilon/4$.

We also need to show that $(H, \tilde{g})$ is a candidate solution. Since H is a candidate subnetwork and $\tilde{g}$ is a candidate flow, we only need to show that $\tilde{g}$ is an ϵ_2-Nash flow in H. Since g is a Nash flow in H, the set of edges $C = \{e : d_e(g_e) \geq B(g)\}$ comprises an $s - t$ cut in H. In fact, for all edges $e \in C$, $d_e(g_e) = B(g)$, if $g_e > 0$, and $d_e(g_e) \geq B(g)$, otherwise. Let us now consider the latency in $\tilde{g}$ of each edge $e \in C$. If $g_e = 0$, then $d_e(\tilde{g}_e) = d_e(g_e) \geq B(g) \geq B(\tilde{g}) - \varepsilon/4$. If $g_e > 0$, then

$$B(\tilde{g}) \geq d_e(\tilde{g}_e) \geq d_e(g_e) - \varepsilon/4 = B(g) - \varepsilon/4 \geq B(\tilde{g}) - \varepsilon/2\,.$$

Therefore, for the flow $\tilde{g}$, we have an $s - t$ cut in H consisting of edges e either with $\tilde{g}_e > 0$ and $B(\tilde{g}) - \varepsilon/2 \leq d_e(\tilde{g}_e) \leq B(\tilde{g})$, or with $\tilde{g}_e = 0$ and $d_e(\tilde{g}_e) \geq B(\tilde{g}) - \varepsilon/4$. By the standard properties of ε-Nash flows (see also in Section 4.1), we obtain that $\tilde{g}$ is a ϵ_2-Nash flow in H.

Hence, we have shown that $(H, \tilde{g})$ is a candidate solution, and that $B(g) \leq B(\tilde{g}) + \varepsilon/4$. Therefore, the algorithm considers both candidate solutions (H, f) and $(H, \tilde{g})$, and returns (H, f), which implies that $B(\tilde{g}) \leq B(f)$. Thus, we obtain (iii), namely that $B(H, 1) = B(g) \leq B(f) + \varepsilon/4$.

To conclude the proof, we next show (iv), namely that $B(f) \leq B(H, 1) + \varepsilon/2$. For the proof, we use the same notation as in (iii). The argument is essentially identical to that used in the second part of the proof of (ii). More specifically, to reach a contradiction, we assume that the candidate flow f, which is an ϵ_2-Nash flow in H, has $B(f) > B(H, 1) + \varepsilon/2$. Then, as before, we should expect that there is a Nash flow f' in H that approximates f and has a bottleneck cost of $B(f') \approx B(f) > B(H, 1)$, a contradiction. Formally, since f is an ϵ_2-Nash flow in H, the set of edges with $d_e(f_e) \geq B(f) - \varepsilon/2$ comprises an $s - t$ cut in H. Then, by the continuity of the latency functions, we can fix a part of the flow routed essentially as in f, so that there is an $s - t$ cut consisting of used edges with latency $B(f) - \varepsilon/2$, and possibly unused edges with latency at least $B(f) - \varepsilon/2$, and reroute the remaining flow on top of it, so that we obtain a Nash flow f' in H. But then, $B(f') \geq B(f) - \varepsilon/2 > B(H, 1)$, which contradicts the definition of the worst equilibrium bottleneck cost $B(H, 1)$ of H. Thus, we obtain (iv), namely that $B(f) \leq B(H, 1) + \varepsilon/2$. □

Therefore, the algorithm of Theorem 5.1 returns a flow-subnetwork pair (H, f) such that f is an $\varepsilon/2$-Nash flow in H, the worst equilibrium bottle-

neck cost of the subnetwork H approximates the worst equilibrium bottleneck cost of H^*, since $B(H^*, 1) \leq B(H, 1) \leq B(H^*, 1) + 5\varepsilon/4$, by (ii) and (iii), and the bottleneck cost of f approximates the worst equilibrium bottleneck cost of H, since $B(H, 1) - \varepsilon/4 \leq B(f) \leq B(H, 1) + \varepsilon/2$, by (iii) and (iv).

Chapter 5

Resolving Braess's Paradox in Random Networks

In this chapter, we study the approximability of the best subnetwork problem for the class of random Erdös-Rényi $\mathcal{G}_{n,p}$ instances proven prone to Braess's paradox by (Roughgarden and Valiant, RSA 2010) and (Chung and Young, WINE 2010). Our main contribution is a polynomial-time approximation-preserving reduction of the best subnetwork problem for such instances to the corresponding problem in a simplified network where all neighbors of s and t are directly connected by 0 latency edges. Building on this, we obtain an approximation scheme that for any constant $\varepsilon > 0$ and with high probability, computes a subnetwork and an ε-Nash flow with maximum latency at most $(1+\varepsilon)L^* + \varepsilon$, where L^* is the equilibrium latency of the best subnetwork. Our approximation scheme runs in polynomial time if the random network has average degree $O(\text{poly}(\ln n))$ and the traffic rate is $O(\text{poly}(\ln \ln n))$, and in quasipolynomial time for average degrees up to $o(n)$ and traffic rates of $O(\text{poly}(\ln n))$.

5.1 Problem-Specific Definitions

We deal with a typical instance $(G(V,E),(d_e)_{e\in E},r)$ of a non atomic CG (*selfish routing* game). In such instances, as noted earlier, all used paths under a Nash flow have a unique minimum latency which we denote as $L(G,d,r)$ or $L(G,r)$, for brevity, assuming that the latency function are given within G.

The paradox seems not an artifact of optimization theory [56, 77], and our motivation is whether in some practically interesting settings, where the paradox occurs, we can efficiently compute a set of edges whose re-

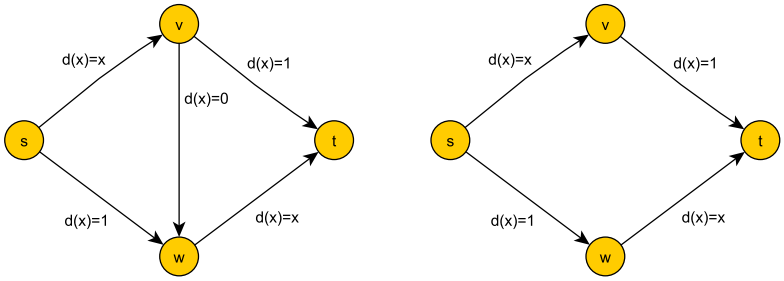

Figure 5.1:
(a) The optimal optimal total latency social cost is $3/2$, achieved by routing half of the flow on each of the paths (s, v, t) and (s, w, t). In the Nash flow, all traffic goes through the path (s, v, w, t) and has selfish cost 2, thus inducing the *worst* PoA = $4/3$ for linear latencies. (b) If we remove the edge (v, w), the Nash flow coincides with the optimal social flow. Hence the network (b) is the *best subnetwork* of network (a) and achieves the *best* possible PoA = 1, with no sacrificed players through slower paths.

moval significantly improves the equilibrium latency.

We only consider linear latencies $d_e(x) = a_e x + b_e$, with $a_e, b_e \geq 0$. We restrict our attention to instances where the coefficients a_e and b_e are randomly selected from a pair of random variables $\mathcal{A}$ and $\mathcal{B}$. Following [24, 84], we say that $\mathcal{A}$ and $\mathcal{B}$ are *reasonable* if:

- $\mathcal{A}$ has bounded range $[A_{\min}, A_{\max}]$ and $\mathcal{B}$ has bounded range $[0, B_{\max}]$, where $A_{\min} > 0$ and $A_{\max}$, $B_{\max}$ are constants, i.e., they do not depend on r and $|V|$.
- There is a closed interval $I_{\mathcal{A}}$ of positive length, such that for every non-trivial subinterval $I' \subseteq I_{\mathcal{A}}$, $\mathbf{Pr}[\mathcal{A} \in I'] > 0$.
- There is a closed interval $I_{\mathcal{B}}$, $0 \in I_{\mathcal{B}}$, of positive length, such that for every non-trivial subinterval $I' \subseteq I_{\mathcal{B}}$, $\mathbf{Pr}[\mathcal{B} \in I'] > 0$. Moreover, for any constant $\eta > 0$, there exists a constant $\delta_\eta > 0$, such that $\mathbf{Pr}[\mathcal{B} \leq \eta] \geq \delta_\eta$.

Flows and Nash Flows. Two flows f and g are *different* if there is an edge e with $f_e \neq g_e$.

Given a flow f, let the latency of f be $L(f) = \max_{p:f_p>0} d_p(f)$. We sometimes write $L_G(f)$ when the network G is not clear from the context.

For an instance $(G(V, E), r)$ and a flow f, we let $E_f = \{e \in E : f_e > 0\}$ be the set of edges used by f, and $G_f(V, E_f)$ be the corresponding subnetwork of G. In a Nash flow f, all players incur a common latency on their

paths, which, according to the previous definitions, is $L(f) = \min_p d_p(f) = \max_{p:f_p>0} d_p(f)$. A Nash flow f on a network $G(V,E)$ is a Nash flow on any subnetwork $G'(V',E')$ of G with $E_f \subseteq E'$.

Every instance (G,r) admits at least one Nash flow, and the players' latency is the same for all Nash flows (see e.g., [77]). For linear latency functions, a Nash flow can be computed efficiently, in strongly polynomial time, while for strictly increasing latencies, the Nash flow is essentially unique (see e.g., [77]).

Best Subnetwork. Recall, the *best subnetwork* H^* of (G,r) is a subnetwork of G with the minimum equilibrium latency, i.e., H^* has $L(H^*,r) \leq L(H,r)$ for any subnetwork H of G. We study the approximability of the *Best Subnetwork Equilibrium Latency* problem, or BestSubEL in short. In BestSubEL, we are given an instance (G,r), and seek for the best subnetwork H^* of (G,r) and its equilibrium latency $L(H^*,r)$.

Good Networks. We restrict our attention to undirected $s-t$ networks $G(V,E)$. We let $n \equiv |V|$ and $m \equiv |E|$. For any vertex v, we let $\Gamma(v) = \{u \in V : \{u,v\} \in E\}$ denote the set of v's neighbors in G. Similarly, for any non-empty $S \subseteq V$, we let $\Gamma(S) = \bigcup_{v\in S} \Gamma(v)$ denote the set of neighbors of the vertices in S, and let $G[S]$ denote the subnetwork of G induced by S. For convenience, we let $V_s \equiv \Gamma(s)$, $E_s \equiv \{\{s,u\} : u \in V_s\}$, $V_t \equiv \Gamma(t)$, $E_t \equiv \{\{v,t\} : v \in V_t\}$, and $V_m \equiv V \setminus (\{s,t\} \cup V_s \cup V_t)$. We also let $n_s = |V_s|$, $n_t = |V_t|$, $n_+ = \max\{n_s,n_t\}$, $n_- = \min\{n_s,n_t\}$, and $n_m = |V_m|$. We sometimes write $V(G)$, $n(G)$, $V_s(G)$, $n_s(G)$, ..., if G is not clear from the context.

It is convenient to think that the network G has a layered structure consisting of s, the set of s's neighbors V_s, an "intermediate" subnetwork connecting the neighbors of s to the neighbors of t, the set of t's neighbors V_t, and t. Then, any $s-t$ path starts at s, visits some $u \in V_s$, proceeds either directly or through some vertices of V_m to some $v \in V_t$, and finally reaches t. Thus, we refer to $G_m \equiv G[V_s \cup V_m \cup V_t]$ as the *intermediate subnetwork* of G. Depending on the structure of G_m, we say that:

- G is a *random* $\mathcal{G}_{n,p}$ network if (i) n_s and n_t follow the binomial distribution with parameters n and p, and (ii) if any edge $\{u,v\}$, with $u \in V_m \cup V_s$ and $v \in V_m \cup V_t$, exists independently with probability p. Namely, the intermediate network G_m is an Erdös-Rényi random graph with $n-2$ vertices and edge probability p, except for the fact that there are no edges in $G[V_s]$ and in $G[V_t]$.

- G is *internally bipartite* if the intermediate network G_m is a bipartite graph with independent sets V_s and V_t. G is *internally complete bipartite* if every neighbor of s is directly connected by an edge to every

neighbor of t.

- G is 0-*latency simplified* if it is internally complete bipartite and every edge e connecting a neighbor of s to a neighbor of t has latency function $d_e(x) = 0$.

The 0-*latency simplification* G_0 of a given network G is a 0-latency simplified network obtained from G by replacing $G[V_m]$ with a set of 0-latency edges directly connecting every neighbor of s to every neighbor of t. Moreover, we say that a 0-latency simplified network G is *balanced*, if $|n_s - n_t| \leq 2n_-$.

We say that a network $G(V, E)$ is (n, p, k)-*good*, for some integer $n \leq |V|$, some probability $p \in (0, 1)$, with $pn = o(n)$, and some constant $k \geq 1$, if G satisfies that:

1. The maximum degree of G is at most $3np/2$, i.e., for any $v \in V$, $|\Gamma(v)| \leq 3np/2$.
2. G is an *expander graph*, namely, for any set $S \subseteq V$, $|\Gamma(S)| \geq \min\{np|S|, n\}/2$.
3. The edges of G have random reasonable latency functions distributed according to $\mathcal{A} \times \mathcal{B}$, and for any constant $\eta > 0$, $\mathbf{Pr}[\mathcal{B} \leq \eta/\ln n]np = \omega(1)$.
4. If $k > 1$ and we randomly partition V_m into k sets $V_m^1, \ldots, V_m^k$, each of cardinality $|V_m|/k$, all the induced subnetworks $G[\{s, t\} \cup V_s \cup V_m^i \cup V_t]$ are $(n/k, p, 1)$-good, with a possible violation of the maximum degree bound by s and t.

If G is a *random* $\mathcal{G}_{n,p}$ network, with n sufficiently large and $p \geq ck \ln n/n$, for some large enough constant $c > 1$, then G is a (n, p, k)-good network with high probability (see e.g., [15]), provided that the latency functions satisfy condition (3) above. Similarly, the random instances considered in [24] are good with high probability. Also note that the 0-latency simplification of a good network is balanced, due to (1) and (2).

5.2 The Approximation Scheme and Outline of the Analysis

In this section, we describe the main steps of the approximation scheme (see also Algorithm 1), and give an outline of its analysis. We let $\varepsilon > 0$ be

Algorithm 1: Approximation Scheme for BestSubEL in Good Networks

Input: Good network $G(V, E)$, rate $r > 0$, approximation guarantee $\varepsilon > 0$

Output: Subnetwork H of G and ε-Nash flow g in H with $L(g) \leq (1+\varepsilon)L(H^*, r) + \varepsilon$

1 if $L(G, r) < \varepsilon$, return G and a Nash flow of (G, r) ;
2 create the 0-latency simplification G_0 of G ;
3 if $r \geq (B_{\max} n_+)/(\varepsilon A_{\min})$, then let $H_0 = G_0$ and let f be a Nash flow of (G_0, r) ;
4 else, let H_0 be the subnetwork and f the $\varepsilon/6$-Nash flow of Thm. 5.7 applied with error $\varepsilon/6$;
5 let H be the subnetwork and let g be the ε-Nash flow of Lemma 5.9 starting from H_0 and f ;
6 return the subnetwork H and the ε-Nash flow g ;

the approximation guarantee, and assume that $L(G, r) \geq \varepsilon$. Otherwise, any Nash flow of (G, r) suffices.

Algorithm 1 is based on an approximation-preserving reduction of BestSubEL for a good network G to BestSubEL for the 0-latency simplification G_0 of G. The first step of our approximation-preserving reduction is to show that the equilibrium latency of the best subnetwork does not increase when we consider the 0-latency simplification G_0 of a network G instead of G itself. Since decreasing the edge latencies (e.g., decreasing $d_{(v,w)}(x) = 1$ to $d_{(v,w)}(x) = 0$ in Fig. 5.1.a) may trigger Braess's paradox, we need Lemma 5.2 and its careful proof to make sure that zeroing out the latency of the intermediate subnetwork does not cause an abrupt increase in the equilibrium latency.

Next, we focus on the 0-latency simplification G_0 of G (step 2 in Alg. 1). We show that if the traffic rate is large enough, i.e., if $r = \Omega(n_+/\varepsilon)$, the paradox has a marginal influence on the equilibrium latency. Thus, any Nash flow of (G_0, r) is an $(1+\varepsilon)$-approximation of BestSubEL (Lemma 5.3, step 4). If $r = O(n_+/\varepsilon)$, we use an approximate version of Caratheodory's theorem (Theorem 5.6) to prove that by an efficient exhaustive search we can obtain an $\varepsilon/6$-approximation of BestSubEL for (G_0, r) (Theorem 5.7, step 4), which in fact we obtain!

We now have a subnetwork H_0 and an $\varepsilon/6$-Nash flow f that comprise a good approximate solution to BestSubEL for the simplified instance (G_0, r). The next step of our approximation-preserving reduction is to extend f to an approximate solution to BestSubEL for the original instance (G, r). The

intuition is that due to the expansion and the reasonable latencies of G, any collection of 0-latency edges of H_0 used by f to route flow from V_s to V_t can be "simulated" by an appropriate collection of low-latency paths of the intermediate subnetwork G_m of G. We first prove this claim for a small part of H_0 consisting only of neighbors of s and neighbors of t with approximately the same latency under f (Lemma 5.8, the proof draws ideas from [24, Lemma 5]). Then, using a careful latency-based grouping of the neighbors of s and of the neighbors of t in H_0, we extend this claim to the entire H_0 (Lemma 5.9). Thus, we obtain a subnetwork H of G and an ε-Nash flow g in H such that $L(g) \leq (1+\varepsilon)L(H^*, r) + \varepsilon$ (step 5).

We summarize our main result. The proof follows by combining Lemma 5.2, Theorem 5.7, and Lemma 5.9 in the way indicated by Algorithm 1 and the discussion above.

Theorem 5.1. Let $G(V, E)$ be (n, p, k)-good network, where $k \geq 1$ is a large enough constant, let $r > 0$ be any traffic rate, and let H^* be the best subnetwork of (G, r). Then, for any $\varepsilon > 0$, Algorithm 1 computes in time $n_+^{O(r^2 A_{\max}^2/\varepsilon^2)}\text{poly}(|V|)$, a flow g and a subnetwork H of G such that with high probability, wrt. the random choice of the latency functions, g is an ε-Nash flow of (H, r) and has $L(g) \leq (1+\varepsilon)L(H^*) + \varepsilon$.

By the definition of reasonable latencies, $A_{\max}$ is a constant. Also, by Lemma 5.3, r affects the running time only if $r = O(n_+/\varepsilon)$. In fact, previous work on selfish network design assumes that $r = O(1)$, see e.g., [77]. Thus, if $r = O(1)$ (or more generally, if $r = O(\text{poly}(\ln\ln n))$) and $pn = O(\text{poly}(\ln n))$, in which case $n_+ = O(\text{poly}(\ln n))$, Theorem 5.1 gives a randomized polynomial-time approximation scheme for BestSubEL in good networks. Moreover, the running time is quasipolynomial for traffic rates up to $O(\text{poly}(\ln n))$ and average degrees up to $o(n)$, i.e., for the entire range of p in [24, 84]. The next sections are devoted to the proofs of Lemmas 5.2 and 5.9, and of Theorem 5.7.

5.3 Network Simplification

We first show that the equilibrium latency of the best subnetwork does not increase when we consider the 0-latency simplification G_0 of a network G instead of G itself.

Lemma 5.2. Let G be any network, let $r > 0$ be any traffic rate, and let H be the best subnetwork of (G, r). Then, there is a subnetwork H' of the 0-latency simplification of H (and thus, a subnetwork of G_0) with $L(H', r) \leq L(H, r)$.

Proof. Throughout the proof, we assume wlog. that all the edges of H are used by the equilibrium flow f of (H, r) (otherwise, we can remove all unused edges from H). The proof is constructive, and at the conceptual level, proceeds in two steps.

For the first step, given the equilibrium flow f of the best subnetwork H of G, we construct a simplification H_1 of H that is internally bipartite and has constant latency edges connecting $\Gamma(s)$ to $\Gamma(t)$. H_1 also admits f as an equilibrium flow, and thus $L(H_1, r) = L(H, r)$. We also show how to further simplify H_1 so that its intermediate bipartite subnetwork becomes acyclic.

To construct the simplification H_1 of H, we let f be the equilibrium flow of H, and let $L \equiv L(H, r)$. For each $u_i \in \Gamma(s)$ and $v_j \in \Gamma(t)$, we let $f_{ij} = \sum_{p=(s,u_i,\ldots,v_j,t)} f_p$ be the flow routed by f from u_i to v_j. The network H_1 is obtained from H by replacing the intermediate subnetwork of H with a bipartite subnetwork connecting $\Gamma(s)$ and $\Gamma(t)$ with constant latency edges. More specifically, instead of the intermediate subnetwork of H, for each $u_i \in \Gamma(s)$ and $v_j \in \Gamma(t)$ with $f_{ij} > 0$, we have an edge $\{u_i, v_j\}$ of constant latency $b_{ij} = L - (a_{\{s,u_i\}} f_{\{s,u_i\}} + b_{\{s,u_i\}}) - (a_{\{v_j,t\}} f_{\{v_j,t\}} + b_{\{v_j,t\}})$ (the corresponding a_{ij} is set to 0). If $f_{ij} = 0$, u_i and v_j are not connected in H_1. We note that by construction, H_1 admits f as an equilibrium flow, and thus $L(H_1, r) = L$.

Furthermore, we modify H_1 by deleting some edges from its intermediate subnetwork so that the induced bipartite subgraph $H_1[\Gamma(s) \cup \Gamma(t)]$ becomes acyclic. Therefore, in the resulting network, for each $u_i \in \Gamma(s)$ and each $v_j \in \Gamma(t)$, there is at most one (s, u_i, v_j, t) path in H_1. Hence, the resulting network admits a unique equilibrium flow with a unique path decomposition.

To this end, let us assume that there is a cycle $C = (u_1, v_2, u_2, \ldots, v_k, u_k, v_1, u_1)$ in the intermediate subnetwork $H_1[\Gamma(s) \cup \Gamma(t)]$. We let $e_{k1} = \{u_k, v_1\}$ be the edge of C with the minimum amount of flow in f, and let f_{k1} be the flow through e_{k1} (see also Fig. 5.2). Then, removing e_{k1}, and updating the flows along the remaining edges of C so that $f'_{ii} = f_{ii} + f_{k1}$, $1 \le i \le k$, and $f'_{i(i+1)} = f_{i(i+1)} - f_{k1}$, $1 \le i \le k-1$, we "break" the cycle C, by eliminating the flow in e_{k1}, and obtain a new equilibrium flow f' of the same rate r and with the same latency L as that of f. Applying this procedure repeatedly to all cycles, we end up with an internally bipartite network H_1 with an acyclic intermediate subnetwork that includes constant latency edges only. Moreover, H_1 admits an equilibrium flow f of latency L. This concludes the first part of the proof.

The second part of the proof is to show that we can either remove some of the intermediate edges of H_1 or zero their latencies, and obtain a subnetwork H' of the 0-latency simplification of H with $L(H', r) \le L(H, r)$. To this end, we describe a procedure where in each step, we either remove

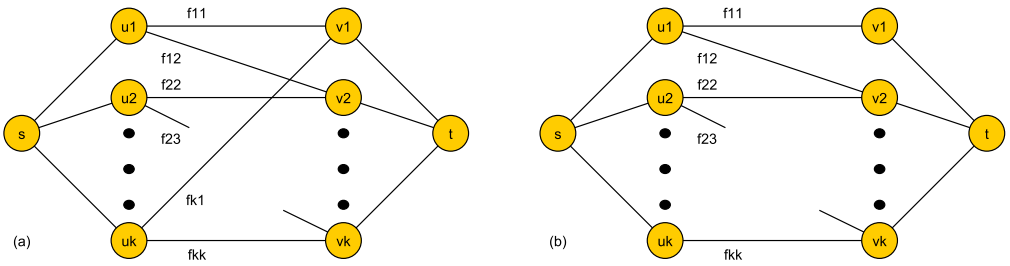

Figure 5.2: In (a), we have a cycle $C = (u_1, v_2, u_2, \ldots, v_k, u_k, v_1, u_1)$ in the intermediate subnetwork $H_1[\Gamma(s) \cup \Gamma(t)]$. We assume that f_{k1} is the minimum amount flow through an edge of C in the equilibrium flow f. In (b), we have removed the edge e_{k1}, and show the corresponding change in the amount of flow on the remaining edges of C. Since the latency functions of the edges in C are constant, the change in the flow does not affect equilibrium.

some intermediate edge of H_1 or zero its latency, without increasing the latency of the equilibrium flow.

Let us focus on an edge $e_{kl} = \{u_k, v_l\}$ with $b_{kl} > 0$, and attempt to set its latency function to $b'_{kl} = 0$. We have also to change the equilibrium flow f to a new flow f' that is an equilibrium flow of latency at most L in the modified network with $b'_{kl} = 0$. We let r_p be the amount of flow moving from an $s - t$ path $p = (s, u_i, v_j, t)$ to the path $p_{kl} = (s, u_k, v_l, t)$ during this change. We note that r_p may be negative, in which case, $|r_p|$ units of flow actually move from p_{kl} to p. Thus, r_p's define a rerouting of f to a new flow f', with $f'_p = f_p - r_p$, for any $s - t$ path p other than p_{kl}, and $f'_{kl} = f_{kl} + \sum_p r_p$.

Next, we show how to compute r_p's so that f' is an equilibrium flow of cost at most L in the modified network (where we want to set $b'_{kl} = 0$). We let $\mathcal{P} = \mathcal{P}_{H_1} \setminus \{p_{kl}\}$ denote the set of all $s - t$ paths in H_1 other than p_{kl}. We let $\vec{F}$ be the $|\mathcal{P}| \times |\mathcal{P}|$ matrix, indexed by the paths $p \in \mathcal{P}$, where $\vec{F}[p_1, p_2] = \sum_{e \in p_1 \cap p_2} a_e - \sum_{e \in p_1 \cap p_{kl}} a_e$, and let $\vec{r}$ be the vector of r_p's. Then, the p-th component of $\vec{F}\vec{r}$ is equal to $d_p(f) - d_p(f')$. In the following, we consider two cases depending on whether $\vec{F}$ is singular or not.

If $\vec{F}$ is non-singular, the linear system $\vec{F}\vec{r} = \varepsilon \vec{1}$ has a unique solution $\vec{r}_\varepsilon$, for any $\varepsilon > 0$. Moreover, due to linearity, for any $a \geq 0$, the unique solution of the system $\vec{F}\vec{r} = a\,\varepsilon \vec{1}$ is $a\,\vec{r}_\varepsilon$. Therefore, for an appropriately small $\varepsilon > 0$, the linear system $\mathcal{Q}_\varepsilon = \{\vec{F}\vec{r} = \varepsilon \vec{1}, f_p - r_p \geq 0 \ \forall p \in \mathcal{P}, f_{kl} + \sum_p r_p \geq 0, d_{p_{kl}}(f') \leq L + b_{kl} - \varepsilon\}$ admits a unique solution $\vec{r}$. We keep increasing ε until one of the inequalities of $\mathcal{Q}_\varepsilon$ becomes tight. If it first becomes $r_p = f_p$ for some path $p = (s, u_i, v_j, t) \in \mathcal{P}$, we remove the edge $\{u_i, v_j\}$ from H_1 and adjust the constant latency of e_{kl} so that $d_{p_{kl}}(f') = L - \varepsilon$. Then, the flow f'

is an equilibrium flow of cost $L-\varepsilon$ for the resulting network, which has one edge less than the original network H_1. If $\sum_p r_p < 0$ and it first becomes $\sum_p r_p = -f_{kl}$, we remove the edge e_{kl} from H_1. Then, f' is an equilibrium flow of cost $L-\varepsilon$ for the resulting network, which again has one edge less than H_1. If $\sum_p r_p > 0$ and it first becomes $d_{p_{kl}}(f') = L + b_{kl} - \varepsilon$, we set the constant latency of the edge e_{kl} to $b'_{kl} = 0$. In this case, f' is an equilibrium flow of cost $L-\varepsilon$ for the resulting network that has one edge of 0 latency more than the initial network H_1.

If $\vec{F}$ is singular, proceeding similarly, we compute r_p's so that f' is an equilibrium flow of cost L in a modified network that includes one edge less than the original network H_1. When $\vec{F}$ if singular, the homogeneous linear system $\vec{F}\vec{r} = \vec{0}$ admits a nontrivial solution $\vec{r} \neq \vec{0}$. Moreover, due to linearity, for any $a \in \mathbb{R}$, $a\vec{r}$ is also a solution to $\vec{F}\vec{r} = \vec{0}$. Therefore, the linear system $\mathcal{Q}_0 = \{\vec{F}\vec{r} = \vec{0}, f_p - r_p \geq 0 \ \forall p \in \mathcal{P}, f_{kl} + \sum_p r_p \geq 0\}$ admits a solution $\vec{r} \neq \vec{0}$ that makes at least one of the inequalities tight. We recall that the p-th component of $\vec{F}\vec{r}$ is equal to $d_p(f) - d_p(f')$. Therefore, for the flow f' obtained from the particular solution $\vec{r}$ of $\mathcal{Q}_0$, the latency of any path $p \in \mathcal{P}$ is equal to L. If $\vec{r}$ is such that $r_p = f_p$ for some path $p = (s, u_i, v_j, t) \in \mathcal{P}$, we remove the edge $\{u_i, v_j\}$ from H_1 and adjust the constant latency of e_{kl} so that $d_{p_{kl}}(f') = L$. Then, the flow f' is an equilibrium flow of cost L for the resulting network, which has one edge less than the original network H_1. If $\vec{r}$ is such that $\sum_p r_p = -f_{kl}$, we remove the edge e_{kl} from H_1. Then, f' is an equilibrium flow of cost L for the resulting network, which again has one edge less than H_1.

Each time we apply the procedure above either we decrease the number of edges of the intermediate network by one or we increase the number of 0-latency edges of the intermediate network by one, without increasing the latency of the equilibrium flow. Moreover, if p_{kl} is disjoint to the paths $p \in \mathcal{P}$, $\vec{F}$ is non-singular (next paragraph) and the procedure above leads to a decrease in the equilibrium latency, and eventually to setting $b'_{kl} = 0$. So by repeatedly applying these steps, we end up with a subnetwork H' of the 0-latency simplification of H with $L(H', r) \leq L(H, r)$.

To show that if p_{kl} is disjoint to the paths $p \in \mathcal{P}$, $\vec{F}$ is non-singular we show that the matrix $\vec{F}$ is positive definite (which implies that $\vec{F}$ is non-singular). We first note that if p_{kl} is disjoint to all $p \in \mathcal{P}$, then for all $p_1, p_2 \in \mathcal{P}$, $\vec{F}[p_1, p_2] = \sum_{e \in p_1 \cap p_2} a_e$. Hence, for all $\vec{x} \in \mathbb{R}^{|\mathcal{P}|}$, $\vec{x}^T\vec{F}\vec{x} = \sum_{e \in E(\mathcal{P})} a_e x_e^2 \geq 0$, where $E(\mathcal{P})$ denotes the set of edges included in the paths of $\mathcal{P}$ and $x_e = \sum_{p: e \in p} x_p$. Since the intermediate network of H_1 is acyclic and any flow in H_1 has a unique path decomposition, if $\vec{x}$ has one or more non-zero components, there is at least one edge e adjacent to either s or t such that $x_e > 0$, and thus $\vec{x}^T\vec{F}\vec{x} > 0$. Otherwise, the difference of the flow

defined by $\vec{x}$ with the trivial flow defined by $\vec{0}$ would indicate the existence of a cycle in the intermediate subnetwork of H_1. This is a contradiction, since by the first part of the proof, the intermediate part of H_1 is acyclic.

□

5.4 Approximating the Best Subnetwork of Simplified Networks

We proceed to show how to approximate the BestSubEL problem in a balanced 0-latency simplified network G_0 with reasonable latencies. We may always regard G_0 as the 0-latency simplification of a good network G. We first prove two useful lemmas (lemmas 5.3 and 5.4) about the maximum traffic rate r up to which BestSubEL remains interesting, and about the maximum amount of flow routed on any edge / path in the best subnetwork.

Lemma 5.3. Let G_0 be any 0-latency simplified network, let $r > 0$, and let H_0^* be the best subnetwork of (G_0, r). For any $\varepsilon > 0$, if $r > \frac{B_{\max} n_+}{A_{\min} \varepsilon}$, then $L(G_0, r) \leq (1 + \varepsilon) L(H_0^*, r)$.

Proof. We first show that for 0-latency simplified instances (G_0, r), we can assume, essentially wlog., that the traffic rate $r = O(n_+/\varepsilon)$. Otherwise, a Nash flow f of (G_0, r) is an $(1 + \varepsilon)$-approximation of the BestSubEL problem in (G_0, r).

To go on with the proof, we assume that $r > \frac{B_{\max} n_+}{A_{\min} \varepsilon}$, let f be a Nash flow of (G_0, r), and consider how f allocates r units of flow to the edges of $E_s \equiv E_s(G_0)$ and to the edges $E_t \equiv E_t(G_0)$. For simplicity, we let $L \equiv L(G_0, r)$ denote the equilibrium latency of G_0, and let $A_s = \sum_{e \in E_s} 1/a_e$ and $A_t = \sum_{e \in E_t} 1/a_e$.

Since G_0 is a 0-latency simplified network and f is a Nash flow of (G_0, r), there are $L_1, L_2 > 0$, with $L_1 + L_2 = L$, such that all used edges incident to s (resp. to t) have latency L_1 (resp. L_2) in the Nash flow f. Since $r > \frac{B_{\max} n_+}{A_{\min}}$, $L_1, L_2 > B_{\max}$ and all edges in $E_s \cup E_t$ are used by f. Moreover, by an averaging argument, we have that there is an edge $e \in E_s$ with $a_e f_e \leq r/A_s$, and that there is an edge $e \in E_t$ with $a_e f_e \leq r/A_t$. Therefore, $L_1 \leq (r/A_s) + B_{\max}$ and $L_2 \leq (r/A_t) + B_{\max}$, and thus, $L \leq \frac{r}{A_s} + \frac{r}{A_t} + 2B_{\max}$.

On the other hand, if we ignore the additive terms b_e of the latency functions, the optimal average latency of the players is $r/A_s + r/A_t$, which implies that $L(H_0^*, r) \geq r/A_s + r/A_t$. Therefore, $L \leq L(H_0^*, r) + 2B_{\max}$. More-

over, since $r > \frac{B_{\max} n_+}{A_{\min}\varepsilon}$, $A_s \le n_s/A_{\min}$, and $A_t \le n_t/A_{\min}$, we have that:

$$\begin{aligned} L(H_0^*, r) &\ge \frac{r}{A_s} + \frac{r}{A_t} \\ &\ge \frac{B_{\max} n_s}{A_{\min}\varepsilon}\frac{A_{\min}}{n_s} + \frac{B_{\max} n_t}{A_{\min}\varepsilon}\frac{A_{\min}}{n_t} \\ &\ge 2B_{\max}/\varepsilon \end{aligned}$$

Therefore, $2B_{\max} \le \varepsilon L(H_0^*, r)$, and $L \le (1+\varepsilon)L(H_0^*, r)$.

□

Lemma 5.4. Let G_0 be a balanced 0-latency simplified network with reasonable latencies, let $r > 0$, and let f be a Nash flow of the best subnetwork of (G_0, r). For any $\varepsilon > 0$, if $\mathbf{Pr}[\mathcal{B} \le \varepsilon/4] \ge \delta$, for some constant $\delta > 0$, there exists a constant $\rho = \frac{24A_{\max}B_{\max}}{\delta\varepsilon A_{\min}^2}$ such that with probability at least $1 - e^{-\delta n_-/8}$, $f_e \le \rho$, for all edges e.

Proof. We proceed to show that in a 0-latency simplified instance (G_0, r), the best subnetwork Nash flow routes $O(r/n_+)$ units of flow on any edge and on any $s-t$ path with high probability (where the probability is with respect to the random choice of the latency function coefficients). Intuitively, we show that in the best subnetwork Nash flow, with high probability, all used edges and all used $s-t$ paths route a volume of flow not significantly larger than their fair share. We first prove the following technical lemma:

Lemma 5.5. Let G_0 be a balanced 0-latency simplified network with reasonable latencies, let $r > 0$ be any traffic rate, and let f be any Nash flow of the best subnetwork of (G_0, r). For any $\varepsilon > 0$, if $L(G, r) \ge \varepsilon$ and $\mathbf{Pr}[\mathcal{B} \le \varepsilon/4] \ge \delta$, for some constant $\delta > 0$, there exists a constant $\gamma = \frac{24A_{\max}}{\delta A_{\min}}$ such that with probability at least $1 - e^{-\delta n_-/8}$, for all edges e, $f_e \le \gamma r/n_+$.

Proof. We let $L \equiv L(G_0, r)$ denote the equilibrium latency and g denote a Nash flow of the original instance (G_0, r). Since G_0 is a 0-latency simplified network and g is a Nash flow of (G_0, r), there are $L_1, L_2 > 0$, with $L_1 + L_2 = L$, such that: (i) for any edge e incident to s, if $b_e < L_1$, $g_e > 0$ and $a_e g_e + b_e = L_1$, while $g_e = 0$, otherwise, and (ii) for any edge e incident to t, if $b_e < L_2$, $g_e > 0$ and $a_e g_e + b_e = L_2$, while $g_e = 0$, otherwise. Namely, all used edges incident to s (resp. to t) have latency L_1 (resp. L_2) in the Nash flow g. Wlog., we assume that $L_1 \ge L_2$, and thus, $L_1 \ge L/2 \ge \varepsilon/2$.

We next show that (i) if $L \ge \varepsilon$ and $\mathbf{Pr}[\mathcal{B} \le \varepsilon/4] \ge \delta$, then with probability at least $1 - e^{-\delta n_-/8}$, $L \le \frac{24A_{\max} r}{\delta n_+}$, and (ii) that for any e, $f_e \le L/A_{\min}$. The lemma follows by combining (i) and (ii).

We start with the proof of (i). Let e be any edge incident to s with $b_e \leq \varepsilon/4$. By the discussion above, in the Nash flow g of (G_0, r), $g_e > 0$ and $a_e g_e + b_e = L_1$. Using that $L_1 \geq L/2 \geq \varepsilon/2$, we obtain that:

$$L_1 = a_e g_e + b_e \leq a_e g_e + \varepsilon/4 \Rightarrow g_e \geq \frac{L_1 - \varepsilon/4}{a_e} \geq \frac{L_1}{2a_e} \geq \frac{L}{4A_{\max}} \tag{5.1}$$

Moreover, since $\mathbf{Pr}[\mathcal{B} \leq \varepsilon/4] \geq \delta$, we use Chernoff bounds (e.g., [48, (7)]), and obtain that:

$$\mathbf{Pr}[|\{e \in E_s(G_0) \text{ with } b_e \leq \varepsilon/4\}| \geq \delta n_s/2] \geq 1 - \mathrm{e}^{-\delta n_s/8} \tag{5.2}$$

Combining (5.2) and (5.1), we obtain that if $L \geq \varepsilon$ and $\mathbf{Pr}[\mathcal{B} \leq \varepsilon/4] \geq \delta$, with probability at least $1-\mathrm{e}^{-\delta n_-/8}$, the flow rate r is at least $\frac{L\delta n_s}{8A_{\max}}$, or equivalently, that:

$$L \leq \frac{8A_{\max} r}{\delta n_s} \leq \frac{24A_{\max} r}{\delta n_+} \tag{5.3}$$

The last inequality holds because G_0 is balanced, and $|n_s - n_t| \leq 2n_-$. This concludes the proof of (i).

To prove (ii), we observe that in the best subnetwork equilibrium flow f, no used edge e has latency greater than L. Therefore, for any used edge e incident to either s or t, we have that:

$$a_e f_e + b_e \leq L \Rightarrow f_e \leq \frac{L}{a_e} \leq \frac{L}{A_{\min}} \tag{5.4}$$

Moreover, any edge e in the intermediate subnetwork of G has $f_e \leq L/A_{\min}$ due to the flow conservation constraints. This concludes the proof of (ii). □

We recall that we always assume that $L(G, r) \geq \varepsilon$, since otherwise the problem of approximating BestSubEL is trivial. Moreover, by the definition of reasonable latency functions, we have that for any constant $\varepsilon > 0$, there is a constant $\delta > 0$, such that $\mathbf{Pr}[\mathcal{B} \leq \varepsilon/4] \geq \delta$. Combining these assumptions with Lemma 5.3 and Lemma 5.5, we obtain Lemma 5.4. So from now on, we can assume, with high probability and wlog., that the Nash flow in the best subnetwork of any simplified instance (G_0, r) routes $O(1)$ units of flow on any used edge and on any used path. □

Approximating the Best Subnetwork of Simplified Networks. First we state an approximate version of Caratheodory's theorem, proved in [12], that is needed for proving the correctness and efficiency of our approximation scheme.

Theorem 5.6 ([12], Theorem 3)**.** Let X be a set of vectors $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ and $\varepsilon > 0$. For every $\mu \in conv(X)$ and $2 \leq p \leq \infty$ there exist an $O(\frac{p\gamma^2}{\varepsilon^2})$ uniform vector $\mu' \in conv(X)$ such that $\|\mu - \mu'\|_p \leq \varepsilon$, where $\gamma = \max_{x \in X} \|x\|_p$ and a *k uniform vector* is a vector that can be written as an average of k vectors of X with replacements allowed.

We proceed to derive an approximation scheme for the best subnetwork of any simplified instance (G_0, r).

Theorem 5.7. Let G_0 be a balanced 0-latency simplified network with reasonable latencies, let $r > 0$, and let H_0^* be the best subnetwork of (G_0, r). Then, for any $\varepsilon > 0$, we can compute, in time $n_+^{O(A_{\max}^2 r^2/\varepsilon^2)}$, a flow f and a subnetwork H_0 consisting of the edges used by f, such that (i) f is an ε-Nash flow of (H_0, r), (ii) $L(f) \leq L(H_0^*, r) + \varepsilon/2$, and (iii) there exists a constant $\rho > 0$, such that $f_e \leq \rho + \varepsilon$, for all e.

Proof. We will use theorem 5.6 to prove the existence of a flow with the properties (i), (ii) and (iii). Then by exhaustive search we will find one such.

For every path $p_{ij} = (s, u_i, v_j, t)$ let $x_{p_{ij}}$ be a vector indexed by the edges of $E(G_0)$, i.e. $x_{p_{ij}} \in \mathbb{R}^{|E(G_0)|}$, that contains everywhere 0, except from the slots that correspond to edges (s, u_i), (u_i, v_j) and (v_j, t) where it contains number r. Clearly, every feasible flow of G_0 can be written as a convex combination of $x_{p_{ij}}$. Let $k = n_+ \cdot n_-$ be the number of different paths in G_0 and for ease of notation let $X = \{x_1, \ldots, x_k\}$ denote the set containing all $x_{p_{ij}}$'s, according to an arbitrary ordering.

Let μ be the Nash flow of the best subnetwork H_0^*. Using theorem 5.6 with set X as defined above, the Nash flow $\mu \in conv(X)$, the $\|\cdot\|_2$ norm, i.e. p=2, and with γ being $\gamma = r\sqrt{3}$ we get that there is an $O(\frac{A_{\max}^2 r^2}{\varepsilon^2})$ uniform vector f such that $\|\mu - f\|_2 \leq \frac{\varepsilon}{4A_{\max}}$, which directly implies $|\mu_e - f_e| \leq \frac{\varepsilon}{4A_{\max}}$ and implies property (iii) for f, due to Lemma 5.4.

The cost of a path $p_{ij} = (s, u_i, v_j, t)$ in H_0^* under f is $d_{p_{ij}}(f) = a_i f_{(s,u_i)} + b_i + a_j f_{(v_j,t)} + b_j$. Because of $|\mu_e - f_e| \leq \frac{\varepsilon}{4A_{\max}}$ we have

$$a_i\mu_{(s,u_i)} + b_i - \frac{\varepsilon}{4} + a_j\mu_{(v_j,t)} + b_j - \frac{\varepsilon}{4} \leq d_{p_{ij}}(f) \leq a_i\mu_{(s,u_i)} + b_i + \frac{\varepsilon}{4} + a_j\mu_{(v_j,t)} + b_j + \frac{\varepsilon}{4},$$

which gives $d_{p_{ij}}(\mu) - \frac{\varepsilon}{2} \leq d_{p_{ij}}(f) \leq d_{p_{ij}}(\mu) + \frac{\varepsilon}{2}$. As μ is the Nash flow of the best subnetwork H_0^* we get

$$L(H_0^*, r) - \frac{\varepsilon}{2} \leq d_p(f) \leq L(H_0^*, r) + \frac{\varepsilon}{2}, \tag{5.5}$$

for any path p in H_0^*.

Let H_0 be the subnetwork induced by the paths used by f. By 5.5, f in H_0 satisfies (i) and (ii) of the theorem.

One can find such an f in time $k^{O(\frac{A_{\max}^2 r^2}{\varepsilon^2})}$ by exhaustively searching for this f in all possible $O(\frac{A_{\max}^2 r^2}{\varepsilon^2})$ combinations of the $k = n_+ \cdot n_-$ paths of G_0. By checking all possible $O(\frac{A_{\max}^2 r^2}{\varepsilon^2})$ uniform vectors for each of these combinations and keeping, among all acceptable flows that satisfy (i) and (ii), the acceptable flow f that minimizes the maximum amount flow routed on any edge, we get $f_e \leq \rho + \varepsilon$, for all edges e, i.e. we get property (iii). The latter is because we know that any Nash flow g of (H_0^*, r) routes $g_e \leq \rho$ units of flow on any edge e (Lemma 5.4), and that in the exhaustive search step, one of the acceptable flows f has $|g_e - f_e| \leq \varepsilon$, for all edges e which implies that there is an acceptable flow f with $f_e \leq \rho + \varepsilon$, for all edges e. □

5.5 Extending the Solution to the Good Network

Given a good instance (G, r), we create the 0-latency simplification G_0 of G, and using Theorem 5.7, we compute a subnetwork H_0 and an $\varepsilon/6$-Nash flow f that comprise an approximate solution to BestSubEL for (G_0, r). Next, we show how to extend f to an approximate solution to BestSubEL for the original instance (G, r). The intuition is that the 0-latency edges of H_0 used by f to route flow from V_s to V_t can be "simulated" by low-latency paths of G_m. We first formalize this intuition for the subnetwork of G induced by the neighbors of s with (almost) the same latency B_s and the neighbors of t with (almost) the same latency B_t, for some B_s, B_t with $B_s + B_t \approx L(f)$. We may think of the networks G and H_0 in the lemma below as some small parts of the original network G and of the actual subnetwork H_0 of G_0. Thus, we obtain the following lemma, which serves as a building block in the proof of Lemma 5.9.

Lemma 5.8. We assume that $G(V, E)$ is a $(n, p, 1)$-good network, with a possible violation of the maximum degree bound by s and t, but with $|V_s|, |V_t| \leq 3knp/2$, for some constant $k > 0$. Also the latencies of the edges in $E_s \cup E_t$ are not random, but there exist constants $B_s, B_t \geq 0$, such that for all $e \in E_s$, $d_e(x) = B_s$, and for all $e \in E_t$, $d_e(x) = B_t$. We let $r > 0$ be any traffic rate, let H_0 be any subnetwork of the 0-latency simplification G_0 of G, and let f be any flow of (H_0, r). We assume that there exists a constant $\rho' > 0$, such that for all $e \in E(H_0)$, $0 < f_e \leq \rho'$. Then, for any $\epsilon_1 > 0$, with high probability, wrt. the random choice of the latency functions of G, we can compute in poly($|V|$) time a subnetwork G' of G, with $E_s(G') = E_s(H_0)$ and $E_t(G') = E_t(H_0)$, and a flow g of (G', r) such that

(i) $g_e = f_e$ for all $e \in E_s(G') \cup E_t(G')$, (ii) g is a $7\epsilon_1$-Nash flow in G', and (iii) $L_{G'}(g) \leq B_s + B_t + 7\epsilon_1$.

Proof. For convenience and wlog., we assume that $E_s(G) = E_s(H_0)$ and that $E_t(G) = E_t(H_0)$, so that we simply write V_s, V_t, E_s, and E_t from now on. For each $e \in E_s \cup E_t$, we let $g_e = f_e$. So, the flow g satisfies (i), by construction.

We compute the extension of g through G_m as an "almost" Nash flow in a modified version of G, where each edge $e \in E_s \cup E_t$ has a capacity $g_e = f_e$ and a constant latency $d_e(x) = B_s$, if $e \in E_s$, and $d_e(x) = B_t$, if $e \in E_t$. All other edges e of G have an infinite capacity and a (randomly chosen) reasonable latency function $d_e(x)$.

We let g be the flow of rate r that respects the capacities of the edges in $E_s \cup E_t$, and minimizes $\text{Pot}(g) = \sum_{e \in E} \int_0^{g_e} d_e(x)dx$. Such a flow g can be computed in strongly polynomial time (see e.g., [85]). The subnetwork G' of G is simply G_g, namely, the subnetwork that includes only the edges used by g. It could have been that g is not a Nash flow of (G, r), due to the capacity constraints on the edges of $E_s \cup E_t$. However, since g is a minimizer $\text{Pot}(g)$, for any $u \in V_s$ and $v \in V_t$, and any pair of $s-t$ paths p, p' going through u and v, if $g_p > 0$, then $d_p(g) \leq d_{p'}(g)$.

We next adjust the proof of [24, Lemma 5], and show that for any $s-t$ path p used by g, $d_p(g) \leq B_s + B_t + 7\epsilon_1$. To prove this, we let $p = (s, u, \ldots, v, t)$ be the $s-t$ path used by g that maximizes $d_p(g)$. We show the existence of a path $p' = (s, u, \ldots, v, t)$ in G of latency $d_{p'}(g) \leq B_s + B_t + 7\epsilon_1$. Therefore, since g is a minimizer of $\text{Pot}(g)$, the latency of the maximum latency g-used path p, and thus the latency of any other g-used $s-t$ path, is at most $B_s + B_t + 7\epsilon_1$, i.e., g satisfies (iii). Moreover, since for any $s-t$ path p, $d_p(g) \geq B_s + B_t$, g is an $7\epsilon_1$-Nash flow in G'.

Let $p = (s, u, \ldots, v, t)$ be the $s-t$ path used by g that maximizes $d_p(g)$. To show the existence of a path $p' = (s, u, \ldots, v, t)$ in G of latency $d_{p'}(g) \leq B_s + B_t + 7\epsilon_1$, we start from $S_0 = \{u\}$ and grow a sequence of vertex sets $S_0 \subseteq S_1 \subseteq \cdots \subseteq S_{i^*}$, stopping when $|\Gamma(S_{i^*})| \geq 3n/5$ for the first time. We use the expansion properties of G, and condition (3), on the distribution of $\mathcal{B}$, in the definition of good networks, and show that these sets grow exponentially fast, and thus, $i^* \leq \ln n$, with high probability. Moreover, we show[1] that there are edges of latency $\epsilon_1 + o(1)$ from $S_0 = \{u\}$ to each vertex of S_1, and edges of latency $\epsilon_1 / \ln n + o(1/\ln n)$ from S_i to each vertex of S_{i+1}, for all $i = 1, \ldots, i^* - 1$. Thus, there is a path of latency at most $2\epsilon_1 + o(1)$ from u to each vertex of S_{i^*}. Similarly, we start from $T_0 = \{v\}$ and grow a

[1]The intuition is that if among the edges e incident to $V_s \cup V_t$, we keep only those with $b_e \leq \epsilon_1$, and among all the remaining edges e, we keep only those with $b_e \leq \epsilon_1 / \ln n$, then due to condition (3) on the distribution of $\mathcal{B}$, a good network G remains an expander.

sequence of vertex sets $T_0 \subseteq T_1 \subseteq \cdots \subseteq T_{j^*}$, stopping when $|\Gamma(T_{j^*})| \geq 3n/5$ for the first time. By exactly the same reasoning, we establish the existence of a path of latency at most $2\epsilon_1 + o(1)$ from each vertex of T_{j^*} to v. Finally, since $|\Gamma(S_{i^*})| \geq 3n/5$ and $|\Gamma(T_{j^*})| \geq 3n/5$, the neighborhoods of S_{i^*} and T_{j^*} contain at least $n/10$ vertices in common. With high probability, most of these vertices can be reached from S_{i^*} and from T_{j^*} using edges of latency $\epsilon_1 + o(1)$. Putting everything together, we find a $u - v$ path (in fact, many of them) of length $O(\ln n)$ and latency at most $6\epsilon_1 + o(1) \leq 7\epsilon_1$.

For completeness, we next give a detailed proof, by adjusting the arguments in the proof of [24, Lemma 5]. For convenience, for each vertex x, we let $d_s(x)$ (resp. $d_t(x)$) be the latency wrt g of the shortest latency path from s to x (resp. from x to t). Also, for any $\delta > 0$, we let $P_b(\delta) \equiv \mathbf{Pr}[\mathcal{B} \leq \delta]$ denote the probability that the additive term of a reasonable latency is at most δ. Recall also that by hypothesis, there exists a constant $\rho' > 0$, such that for all $e \in E(H_0)$, $f_e \leq \rho'$. Hence, the total flow through G (and through H_0) is $r \leq \rho' n_+$.

At the conceptual level, the proof proceeds as explained above. We start with $S_0 = \{u\}$. By hypothesis, the flow entering u is at most ρ'. By the expansion property of good networks and by Chernoff bounds[2], with high probability, there are at least $P_b(\epsilon_1)np/4$ edges e adjacent to u with $b_e \leq \epsilon_1$. At most half of these edges have flow greater than $\frac{8\rho'}{P_b(\epsilon_1)np}$, thus there are at least $P_b(\epsilon_1)np/8$ edges adjacent to u with latency, wrt g, less than $\frac{8A_{\max}\rho'}{P_b(\epsilon_1)np} + \epsilon_1$. We now let $d_1 = B_s + \frac{8A_{\max}\rho'}{P_b(\epsilon_1)np} + \epsilon_1$ and $S_1 = \{x \in V : d_s(x) \leq d_1\}$. By the discussion above, $|S_1| \geq P_b(\epsilon_1)np/8$.

We now inductively define a sequence of vertex sets S_i and upper bounds d_i on the latency of the vertices in S_i from s, such that $S_i \subseteq S_{i+1}$ and $d_i < d_{i+1}$. This sequence stops the first time that $|\Gamma(S_i)| \geq 3n/5$. We inductively assume that the vertex set S_i and the upper bound d_i on the latency of the vertices in S_i are defined, and that $|\Gamma(S_i)| < 3n/5$. By the expansion property of good networks $|\Gamma(S_i) \setminus S_i| \geq np|S_i|/3$, for sufficiently large n. Thus, with probability at least $1 - e^{P_b(\epsilon_1/\ln n)np|S_i|/24}$, there are at least $P_b(\frac{\epsilon_1}{\ln n})np|S_i|/6$ vertices outside S_i that are connected to a vertex in S_i by an edge e with $b_e \leq \epsilon_1/\ln n$. Let S_i' be the set of such vertices, and let E_i be the set of edges that for each vertex $v \in S_i'$, includes a unique edge $e \in E_i$ with $b_e \leq \epsilon_1/\ln n$ connecting v to a vertex in S_i. Since the flow g may be assumed to be acyclic, a volume $r \leq \rho' n_+$ of flow is routed through the cut $(S_i, V \setminus S_i)$. Then, at most half of the edges in E_i have flow greater than

[2]We repeatedly use the following form of the Chernoff bound (see e.g., [48]): Let $X_1, \ldots, X_k$ be random variables independently distributed in $\{0, 1\}$, and let $X = \sum_{i=1}^{k} X_i$. Then, for all $\epsilon \in (0, 1)$, $\mathbf{Pr}[X < (1 - \epsilon)\mathbf{E}[X]] \leq e^{-\epsilon^2 \mathbf{E}[X]/2}$, where e is the basis of natural logarithms.

$2\rho' n_+ / |S'_i|$. Consequently, at least half of the vertices $v \in S'_i$ have latency from s:

$$\begin{aligned} d_s(x) &\leq d_i + \frac{\epsilon_1}{\ln n} + A_{\max} \frac{2\rho' n_+}{|S'_i|} \\ &\leq d_i + \frac{\epsilon_1}{\ln n} + \frac{12 A_{\max} \rho' n_+}{P_b(\frac{\epsilon_1}{\ln n}) np |S_i|} \end{aligned}$$

Thus, we define the next latency upper bound d_{i+1} in the sequence as:

$$d_{i+1} = d_i + \frac{\epsilon_1}{\ln n} + \frac{12 A_{\max} \rho' n_+}{P_b(\frac{\epsilon_1}{\ln n}) np |S_i|},$$

and we let $S_{i+1} = \{x \in V(G) | d_s(x) \leq d_{i+1}\}$. By the discussion above, and using the inductive definition of S_i's, we obtain that:

$$\begin{aligned} |S_{i+1}| &\geq \left(\tfrac{1}{12} P_b(\epsilon_1 / \ln n) np + 1\right) |S_i| \\ &\geq \left(\tfrac{1}{12} P_b(\epsilon_1 / \ln n) np + 1\right)^i |S_1| \end{aligned}$$

We recall that i^* is the first index i such that $|\Gamma(S_i)| \geq 3n/5$. Then, the inequality above implies that:

$$i^* \leq \frac{\ln(3n/(5|S_1|))}{\ln\left(\frac{1}{12} P_b(\epsilon_1 / \ln n) np + 1\right)} \leq \frac{\ln(24n/(5P_b(\epsilon_1) np))}{\ln\left(\frac{1}{12} P_b(\epsilon_1 / \ln n) np + 1\right)}$$

Using that $pn \geq \ln n$ and that $P_b(\epsilon_1 / \ln n) np = \omega(1)$, the inequality above implies that $i^* \leq \ln n$, for sufficiently large n.

Therefore, we obtain an upper bound on the latency from s of any vertex in S_{i^*} :

$$\begin{aligned} d_{i^*} &\leq d_0 + i^* \frac{\epsilon_1}{\ln n} + \sum_{i=1}^{i^*} \frac{12 A_{\max} \rho' n_+}{P_b(\frac{\epsilon_1}{\ln n}) np |S_i|} \\ &\leq d_1 + \frac{\epsilon_1}{\ln n} \ln n + \sum_{i=1}^{\ln n} \frac{12 A_{\max} \rho' n_+}{P_b(\frac{\epsilon_1}{\ln n}) np \left(\frac{1}{12} P_b(\frac{\epsilon_1}{\ln n}) np + 1\right)^i |S_1|} \\ &= d_1 + \epsilon_1 + \frac{12 A_{\max} \rho' n_+}{P_b(\frac{\epsilon_1}{\ln n}) np |S_1|} \sum_{i=1}^{\ln n} \left(\tfrac{1}{12} P_b(\tfrac{\epsilon_1}{\ln n}) np + 1\right)^{-i} \\ &\leq \left(B_s + \frac{8 A_{\max} \rho'}{P_b(\epsilon_1) np} + \epsilon_1\right) + \epsilon_1 + \frac{96 A_{\max} \rho' n_+}{P_b(\frac{\epsilon_1}{\ln n}) P_b(\epsilon_1) (np)^2} \sum_{i=1}^{\infty} 2^{-i} \\ &\leq B_s + 2\epsilon_1 + \frac{8 A_{\max} \rho'}{P_b(\epsilon_1) np} + \frac{144 A_{\max} \rho' k}{P_b(\frac{\epsilon_1}{\ln n}) P_b(\epsilon_1) np} \end{aligned}$$

For the penultimate inequality, we use that $P_b(\epsilon_1/\ln n)np = \omega(1)$, which implies that $1 + P_b(\epsilon_1/\ln n)np/12 \geq 2$, for n sufficiently large. For the last inequality, we use that $n_+ \leq 3knp/2$, for some constant $k > 0$, by hypothesis.

Moreover, we observe that probability that the above construction fails is at most:

$$\begin{aligned}\sum_{i=1}^{i^*} \mathrm{e}^{-P_b(\epsilon_1/\ln n)np|S_i|/24} &\leq \sum_{i=1}^{i^*} \mathrm{e}^{-\left(\frac{1}{12}P_b(\epsilon_1/\ln n)np+1\right)^i|S_1|/24} \\ &\leq \ln n\, \mathrm{e}^{-\left(\frac{1}{12}P_b(\epsilon_1/\ln n)np+1\right)P_b(\epsilon_1)np/192}\end{aligned}$$

Therefore, the construction above succeeds with high probability.

Similarly, we start from $T_0 = \{v\}$, and inductively define a sequence of vertex sets $T_0 \subseteq T_1 \subseteq \cdots \subseteq T_{j^*}$, and a sequence of upper bounds $d'_0 < d'_1 < \cdots < d'_{j^*}$ on the latency from t of the vertices in each T_j. We let $T_j = \{x \in V(G) | d_t(x) \leq d'_j\}$. The sequence stops as soon as $|\Gamma(T_j)| \geq 3n/5$ for the first time. Namely, j^* is the first index with $|\Gamma(T_{j^*})| \geq 3n/5$. Using exactly the same arguments, we can show that with high probability, we have that $j^* \leq \ln n$, and that:

$$d'_{j^*} \leq B_t + 2\epsilon_1 + \frac{8A_{\max}\rho'}{P_b(\epsilon_1)np} + \frac{144A_{\max}\rho' k}{P_b(\frac{\epsilon_1}{\ln n})P_b(\epsilon_1)np}$$

Wlog., we assume that $S_{i^*} \cap T_{j^*} = \emptyset$. Since $|\Gamma(S_{i^*})| + |\Gamma(T_{j^*})| \geq 6n/5$, there are at least $n/10$ edge disjoint paths of length at most 2 between S_{i^*} and T_{j^*}. Furthermore, by Chernoff bounds, with high probability, there are at least $P_b(\epsilon_1)^2 n/12$ such paths with both edges e on the path having $b_e \leq \epsilon_1$. At most half of these paths have flow more than $2\frac{12\rho' n_+}{P_b(\epsilon_1)^2 n}$ and thus there is a path from a vertex of S_{i^*} to a vertex of T_{j^*} that costs at most $2\epsilon_1 + 2A_{\max}\frac{24\rho' n_+}{P_b(\epsilon_1)^2 n}$.

Putting everything together, we have that there is a path p' that starts from s, moves to u, goes through vertices of the sequence $S_1, \ldots, S_{i^*}$, proceeds to a vertex of $\Gamma(S_{i^*}) \cap \Gamma(T_{j^*})$, and from there, continues through vertices of the sequence $T_{j^*}, \ldots, T_1$, until finally reaches v, and then t. The latency of this path is:

$$d_{p'}(g) \leq B_s + B_t + 6\epsilon_1 + 2\Big(\frac{8A_{\max}\rho'}{P_b(\epsilon_1)np} + \frac{48A_{\max}\rho' k}{P_b(\frac{\epsilon_1}{\ln n})P_b(\epsilon_1)np}\Big) + \frac{48A_{\max}\rho' n_+}{P_b(\epsilon_1)^2 n}$$

We recall that since the flow g is a the minimizer of $\mathrm{Pot}(g)$, for any g-used path $p = (s, u, \ldots, v, t)$, $d_p(g) \leq d_{p'}(g)$. Thus we obtain that any

g-used path $p = (s, u, \ldots, v, t)$ has latency

$$d_p(g) \leq B_s + B_t + 6\epsilon_1 + 2\Big(\frac{8A_{\max}\rho'}{P_b(\epsilon_1)np} + \frac{48A_{\max}\rho' k}{P_b(\frac{\epsilon_1}{\ln n})P_b(\epsilon_1)np}\Big) + \frac{48A_{\max}\rho' n_+}{P_b(\epsilon_1)^2 n}$$

Using the hypothesis that $n_+ \leq 3knp/2$, for constant $k > 0$, and that $P_b(\epsilon_1/\ln n)np = \omega(1)$, which is condition (3), in the definition of good networks, we obtain that for any constant $\epsilon_1 > 0$, $d_p(g) \leq B_s + B_t + 7\epsilon_1$, for sufficiently large n. □

Grouping the Neighbors of s and t. Let us now consider the entire network G and the entire subnetwork H_0 of G_0. Lemma 5.8 can be applied only to subsets of edges in $E_s(H_0)$ and in $E_t(H_0)$ that have (almost) the same latency under f. Hence, we partition the neighbors of s and the neighbors of t into classes V_s^i and V_t^j according to their latency. For convenience, we let $\epsilon_2 = \varepsilon/6$, i.e., f is an ϵ_2-Nash flow, and $L \equiv L_{H_0}(f)$. By Theorem 5.7, applied with error $\epsilon_2 = \varepsilon/6$, there exists a ρ such that for all $e \in E(H_0)$, $0 < f_e \leq \rho + \epsilon_2$. Hence, $L \leq 2A_{\max}(\rho + \epsilon_2) + 2B_{\max}$ is bounded by a constant.

We partition the interval $[0, L]$ into $\kappa = \lceil L/\epsilon_2 \rceil$ subintervals, where the i-th subinterval is $I^i = (i\epsilon_2, (i+1)\epsilon_2]$, $i = 0, \ldots, \kappa - 1$. We partition the vertices of V_s (resp. of V_t) that receive positive flow by f into κ classes V_s^i (resp. V_t^i), $i = 0, \ldots, \kappa - 1$. Precisely, a vertex $x \in V_s$ (resp. $x \in V_t$), connected to s (resp. to t) by the edge $e_x = \{s, x\}$ (resp. $e_x = \{x, t\}$), is in the class V_s^i (resp. in the class V_t^i), if $d_{e_x}(f_{e_x}) \in I_i$. If a vertex $x \in V_s$ (resp. $x \in V_t$) does not receive any flow from f, x is removed from G and does not belong to any class. Hence, from now on, we assume that all neighbors of s and t receive positive flow from f, and that $V_s^0, \ldots V_s^{\kappa-1}$ (resp. $V_t^0, \ldots, V_t^{\kappa-1}$) is a partitioning of V_s (resp. V_t). In exactly the same way, we partition the edges of E_s (resp. of E_t) used by f into k classes E_s^i (resp. E_t^i), $i = 0, \ldots, \kappa - 1$.

To find out which parts of the subnetwork H_0 will be connected through the intermediate subnetwork of G, using the construction of Lemma 5.8, we further classify the vertices of V_s^i and V_t^j based on the neighbors of t and on the neighbors of s, respectively, to which they are connected by f-used edges in the subnetwork H_0. In particular, a vertex $u \in V_s^i$ belongs to the classes $V_s^{(i,j)}$, for all $j \in \{0, \ldots, \kappa - 1\}$ such that there is a vertex $v \in V_t^j$ with $f_{\{u,v\}} > 0$. Similarly, a vertex $v \in V_t^j$ belongs to the classes $V_t^{(i,j)}$, for all $i \in \{0, \ldots, \kappa - 1\}$ such that there is a vertex $u \in V_s^i$ with $f_{\{u,v\}} > 0$. We note that a vertex $u \in V_s^i$ (resp. $v \in V_t^j$) may belong to many different classes $V_s^{(i,j)}$ (resp. to $V_t^{(i,j)}$), and that the class $V_s^{(i,j)}$ is non-empty iff the class $V_t^{(i,j)}$ is non-empty, i.e., non-empty classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$ appear in pairs. We let $k \leq \kappa^2$ be the number of pairs (i, j) for which $V_s^{(i,j)}$ and $V_t^{(i,j)}$ are non-empty. We note that k is a constant, i.e., does not depend on $|V|$ and r. We let $E_s^{(i,j)}$

be the set of edges connecting s to the vertices in $V_s^{(i,j)}$ and $E_t^{(i,j)}$ be the set of edges connecting t to the vertices in $V_t^{(i,j)}$.

Building the Intermediate Subnetworks of G. The last step is to replace the 0-latency simplified parts connecting the vertices of each pair of classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$ in H_0 with a subnetwork of G_m. To this end, we randomly partition the set V_m of intermediate vertices of G into k subsets, each of cardinality (roughly) $|V_m|/k$, and associate a different such subset $V_m^{(i,j)}$ with any pair of non-empty classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$. For each pair (i,j) for which the classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$ are non-empty, we consider the induced subnetwork $G^{(i,j)} \equiv G[\{s,t\} \cup V_s^{(i,j)} \cup V_m^{(i,j)} \cup V_t^{(i,j)}]$, which is a $(n/k, p, 1)$-good network, by condition (4) in the definition of good networks, and because G is a (n,p,k)-good network. Therefore, we can apply Lemma 5.8 to $G^{(i,j)}$, with $H_0^{(i,j)} \equiv H_0[\{s,t\} \cup V_s^{(i,j)} \cup V_t^{(i,j)}]$ in the role of H_0, the restriction $f^{(i,j)}$ of f to $H_0^{(i,j)}$ in the role of the flow f, and $\rho' = \rho + \epsilon_2$. Moreover, we let $B_s^{(i,j)} = \max_{e \in E_s^{(i,j)}} d_e(f_e)$ and $B_t^{(i,j)} = \max_{e \in E_t^{(i,j)}} d_e(f_e)$ correspond to B_s and B_t, and introduce constant latencies $d'_e(x) = B_s^{(i,j)}$ for all $e \in E_s^{(i,j)}$ and $d'_e(x) = B_t^{(i,j)}$ for all $e \in E_t^{(i,j)}$, as required by Lemma 5.8. Thus, we obtain, with high probability, a subnetwork $H^{(i,j)}$ of $G^{(i,j)}$ and a flow $g^{(i,j)}$ that routes as much flow as $f^{(i,j)}$ on all edges of $E_s^{(i,j)} \cup E_t^{(i,j)}$, and satisfies the conclusion of Lemma 5.8, if we keep in $H^{(i,j)}$ the constant latencies $d'_e(x)$ for all $e \in E_s^{(i,j)} \cup E_t^{(i,j)}$.

The final outcome is the union of the subnetworks $H^{(i,j)}$, denoted H (H has the latency functions of the original instance G), and the union of the flows $g^{(i,j)}$, denoted g, where the union is taken over all k pairs (i,j) for which the classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$ are non-empty. By construction, all edges of H are used by g. We obtain lemma 5.9 by showing that if $\epsilon_1 = \varepsilon/42$ and $\epsilon_2 = \varepsilon/6$, the flow g is an ε-Nash flow of (H, r), and satisfies $L_H(g) \leq L_{H_0}(f) + \varepsilon/2$.

Lemma 5.9. Let any $\varepsilon > 0$, let $k = \lceil 12(A_{\max}(\rho + \varepsilon) + B_{\max})/\varepsilon \rceil^2$, let $G(V,E)$ be an (n,p,k)-good network, let $r > 0$, let H_0 be any subnetwork of the 0-latency simplification of G, and let f be an $(\varepsilon/6)$-Nash flow of (H_0, r) for which there exists a constant $\rho' > 0$, such that for all $e \in E(H_0)$, $0 < f_e \leq \rho'$. Then, with high probability, wrt. the random choice of the latency functions of G, we can compute in poly($|V|$) time a subnetwork H of G and an ε-Nash flow g of (H, r) with $L_H(g) \leq L_{H_0}(f) + \varepsilon/2$.

Proof. We consider the subnetwork H (with the original latency functions of G), computed as the union of subnetworks $H^{(i,j)}$, and the flow g, computed as the union of the flows $g^{(i,j)}$, where the union is taken over all k pairs (i,j) for which the classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$ are non-empty. We recall that by construction, all edges of H are used by g. We show that if $\epsilon_1 = \varepsilon/42$

and $\epsilon_2 = \varepsilon/6$, the flow g is an ε-Nash flow of (H, r), and satisfies $L_H(g) \leq L_{H_0}(f) + \varepsilon/2$. We stress that the edge and path latencies here are calculated with respect to the original latency functions of G and under the edge congestion induced by the flow g (or the flow f).

For convenience, we let $B^{(i,j)} = B_s^{(i,j)} + B_t^{(i,j)}$ for any pair of non-empty classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$. Since the difference in the latency of any edges in the same group is at most ϵ_2, we obtain that for any edge $e \in E_s^{(i,j)}$, $B_s^{(i,j)} - \epsilon_2 \leq d_e(f_e) \leq B_s^{(i,j)}$, and similarly, that for any edge $e \in E_t^{(i,j)}$, $B_t^{(i,j)} - \epsilon_2 \leq d_e(f_e) \leq B_t^{(i,j)}$. Therefore, since H_0 is a 0-latency simplified network, and since by hypothesis, all the edges of H_0 are used by f, for any pair of non-empty classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$, and for any $s-t$ path p going through a vertex of $V_s^{(i,j)}$ and a vertex of $V_t^{(i,j)}$,

$$B^{(i,j)} - 2\epsilon_2 \leq d_p(f) \leq B^{(i,j)}$$

Moreover, since f is an ϵ_2-Nash flow of (H_0, r), for any $s-t$ path $p \in \mathcal{P}_{H_0}$,

$$L_{H_0}(f) - \epsilon_2 \leq d_p(f) \leq L_{H_0}(f)$$

Combining the two inequalities above, we obtain that for any pair of non-empty classes $V_s^{(i,j)}$ and $V_t^{(i,j)}$,

$$B^{(i,j)} - 2\epsilon_2 \leq L_{H_0}(f) \leq B^{(i,j)} + \epsilon_2 \tag{5.6}$$

As for the flow g, by construction, we have that $g_e = f_e$ for all edges $e \in E_s \cup E_t$. Therefore, for any edge $e \in E_s^{(i,j)}$, $B_s^{(i,j)} - \epsilon_2 \leq d_e(g_e) \leq B_s^{(i,j)}$, and similarly, for any edge $e \in E_t^{(i,j)}$, $B_t^{(i,j)} - \epsilon_2 \leq d_e(g_e) \leq B_t^{(i,j)}$. Thus, by Lemma 5.8, and since all the edges of any subnetwork $H^{(i,j)}$ are used by g, for any $s-t$ path p in the subnetwork $H^{(i,j)}$, $B^{(i,j)} - 2\epsilon_2 \leq d_p(g) \leq B^{(i,j)} + 7\epsilon_1$. Using that (5.6), we obtain that for any subnetwork $H^{(i,j)}$ and any $s-t$ path p of $H^{(i,j)}$,

$$L_{H_0}(f) - 3\epsilon_2 \leq d_p(g) \leq L_{H_0}(f) + 2\epsilon_2 + 7\epsilon_1 \tag{5.7}$$

Furthermore, we recall that the subnetworks $H^{(i,j)}$ only have in common the vertices s and t, and possibly some vertices of $V_s \cup V_t$ and some edges of $E_s \cup E_t$. They have neither any other vertices in common, nor any edges connecting vertices in the intermediate parts of different subnetworks $H^{(i,j)}$ and $H^{(i',j')}$. Hence, any $s-t$ path p of H passes through a single subnetwork $H^{(i,j)}$. Therefore, and since by construction, all the edges and the paths of H are used by g, (5.7) holds for any $s-t$ path p of H.

Thus, we have shown that g is a $(5\epsilon_2 + 7\epsilon_1)$-Nash flow of (H, r), and that $L_H(g) \leq L_{H_0}(f) + 2\epsilon_2 + 7\epsilon_1$. Using $\epsilon_2 = \varepsilon/6$ and $\epsilon_1 = \varepsilon/42$, we obtain the performance guarantees of g as stated in Lemma 5.9. □

Chapter 6

Congestion Games with Risk Averse Players

Congestion games ignore the stochastic nature of resource delays and the risk-averse attitude of the players to uncertainty. To take these aspects into account, we introduce two variants of atomic congestion games, one with *stochastic players*, where each player assigns load to her strategy independently with a given probability, and another with *stochastic edges*, where the latency functions are random. In both variants, the players are risk-averse, and their individual cost is a player-specific quantile of their delay distribution. We focus on parallel-link networks and investigate how the main properties of stochastic congestion games depend on the risk attitude and the participation probabilities of the players. In a nutshell, we prove that stochastic congestion games on parallel-links admit an efficiently computable pure Nash equilibrium if the players have either the same risk attitude or the same participation probabilities, and also admit a potential function if the players have the same risk attitude. On the negative side, we present examples of stochastic games with players of different risk attitudes that do not admit a potential function. As for the inefficiency of equilibria, for parallel-link networks with linear delays, we prove that the Price of Anarchy is $\Theta(n)$, where n is the number of stochastic players, and may be unbounded, in case of stochastic edges.

6.1 Introducing the Models

In this chapter we generalize atomic congestion games. Recall that a (standard) atomic CG is the tuple $\mathcal{G}(N, E, (S_i)_{i\in N}, (d_e)_{e\in E})$. On the corresponding sections we will get more specific on the generalization we do, by explicitly

defining the generalizing models.

In the technical part, we restrict our attention to symmetric congestion games on parallel-link networks, where strategies are singletons and there is a strategy for every resource. Since we mostly consider networks, we use the terms "resource" and "edge" and "strategy" and "path" interchangeably.

6.2 Congestion Games with Stochastic Players

6.2.1 The Model

In Congestion Games with Stochastic Players, each player i is described by a tuple (p_i, δ_i), where $p_i \in [0, 1]$ is the probability that player i participates in the game, by assigning a unit of load to her strategy, and $\delta_i \in \left[\frac{1}{2}, 1\right]$ is the *confidence level* (or risk-aversion) of player i. Essentially, each player i is associated with a Bernoulli random variable X_i that is 1 with probability p_i, and 0 with probability $1-p_i$. Then, the load of each edge e in a configuration s is the random variable $N_e(s) = \sum_{i:e\in s_i} X_i$, and the cost of a strategy q in s is the random variable $D_q(s) = \sum_{e\in q} d_e(N_e(s))$.

Given that player i participates in the game, the delay of player i in s is given by the random variable:

$$D_i(s) = \sum_{e\in s_i} d_e\left(1 + \sum_{j\neq i:\ e\in s_j} X_j\right).$$

Note that when $X_i = 1$, $D_i(s) = D_{s_i}(s)$.

The (risk-averse) individual cost $c_i(s)$ perceived by player i in s is the δ_i-quantile (or value-at-risk) of $D_i(s)$. Formally, $c_i(s) = \min\{t : \mathbf{Pr}[D_i(s) \leq t] \geq \delta_i\}$. We note that for parallel-link networks, the (risk-averse) individual cost of the players can be computed efficiently. PNE are defined as before, but with respect to the risk-averse individual cost of the players.

Depending on whether players have the same participation probabilities p_i and/or the same confidence levels δ_i, we distinguish between four classes of congestion games with stochastic players:

- *homogeneous*, where all players have the same participation probability p and confidence level δ.
- *p-homogeneous*, where all players have the same participation probability p, but may have different confidence levels.

- *δ-homogeneous*, where all players have the same confidence level δ, but may have different participation probabilities.
- *heterogeneous*, where both the participation probabilities and the confidence levels may be different.

6.2.2 Stochastic Players on Parallel Links: Existence and Computation of PNE

In the following, we restrict ourselves to Congestion Games with Stochastic Players on parallel-link networks, and investigate the existence and the efficient computation of PNE for the four cases considered above.

Homogeneous Stochastic Players. If the players are homogeneous, stochastic congestion games on parallel-links are equivalent to standard congestion games on parallel-links (but with possibly different latencies), because the (risk-averse) individual cost of each player in a configuration s depends only on the link e and its congestion s_e.

Theorem 6.1. Congestion Games with Homogeneous Stochastic Players on parallel-link networks admit an exact potential function. Moreover, a pure Nash equilibrium can be computed in polynomial time.

Proof. The existence of a potential function implies the existence of equilibrium. Rosenthal's potential function ([73]) suites this case as players, under any configuration, perceive exactly the same cost on any edge e, a cost that depends only on the number of players on e.

Formally, let p and δ denote the common p_i's and δ_i's of the players, and for each edge e, define a function $f_e : \mathbb{N} \to \mathbb{R}$ with

$$f_e(r) = \begin{cases} 0, & \text{if } r = 0 \\ \min\left\{t : \Pr\left[d_e(1 + \sum_{i=1}^{r-1} Y_i) \le t\right] \ge \delta\right\}, & \text{if } r > 0. \end{cases}$$

where for all $i \in [r]$, Y_i is a Bernoulli random variable (independent of others) with probability of success p. Observe that for any configuration s and any player i with $s_i = e$, we have $c_i(s) = f_e(s_e)$.

Defining

$$\Phi(s) = \sum_e \sum_{i=1}^{s_e} f_e(i),$$

it is easy to verify that it is an exact potential function as for any player i and any two configurations s, s' such that $s_{-i} = s'_{-i}$, $s_i = e$ and $s'_i = e'$

$$\Phi(s) - \Phi(s') = f_e(s_e) - f_{e'}(s_{e'} + 1) = c_i(s) - c_i(s').$$

In order to polynomially compute a PNE, we can simply use a *Greedy Best Response* algorithm. We insert the players in the game, one by one, and the player just inserted does a best response move. Each such move can be computed in time $O(m + i^2)$, where i is the number of players in the game so far, as for each edge e, the cumulative distribution function of $D_e(s)$ can be computed via dynamic programming. One can easily verify, by induction, that on any round of this procedure, the players are in a PNE. Consequently, we get a PNE when the procedure terminates. Using memoization to avoid recalculations, the total time needed is $O(n \cdot m + n^2)$. □

p-Homogeneous Stochastic Players. In this case, a stochastic game is equivalent to a congestion game on parallel links with player-specific payoffs [63], as the (risk-averse) individual cost of each player in a configuration s depends only on the link e, its congestion s_e, and i's confidence level δ_i. Thus, we obtain:

Corollary 6.2. Congestion Games with p-Homogeneous Stochastic Players on parallel-link networks admit a PNE. Moreover, a PNE can be computed in polynomial time.

Milchtaich [63] proved that a parallel-link congestion game with general player-specific payoffs may not admit a potential function. In our case however, the players' individual costs are correlated with each other, as for any edge, there is a common distribution on which they depend. Nevertheless, we next show that parallel-link games with p-homogeneous stochastic players and linear latencies may not admit a potential function.

Theorem 6.3. There are Congestion Games with p-Homogeneous Stochastic Players on parallel-link networks with linear delays that do not admit a potential function.

Proof. It suffices to show that there is an infinite improvement cycle, i.e. an infinite sequence of deviations for which each deviating player perceives less cost. We will modify Milchtaich's counter-example ([63]) to fit our case. Since players have the same probability of participation p, the load on each edge e that player i considers is a binomial distribution: $\sum_{j \neq i:\ e \in s_j} X_j$.

Fix $p = 0.75$, and consider three edges, e_1, e_2 and e_3, and three players that will deviate, with $\delta_1 = 0.75$, $\delta_2 = 0.58$ and $\delta_3 = 0.6$. Also, assume that there are $n_1 = 25$ extra players on e_1, $n_2 = 20$ extra players on e_2 and $n_3 = 9$ extra players on e_3. The latency functions of the edges are: $f_1(k) = 3k + 71$, $f_2(k) = 6k + 33$ and $f_3(k) = 15k + 1$.

In the following, an infinite better response cycle is described, consisting of six different configurations that interchangeably follow one another:

	e_1	e_2	e_3
i	$\{1,2\}$	$\{3\}$	
ii	$\{1,2\}$		$\{3\}$
iii	$\{2\}$		$\{1,3\}$
iv		$\{2\}$	$\{1,3\}$
v		$\{2,3\}$	$\{1\}$
vi	$\{1\}$	$\{2,3\}$	
	$\{1,2\}$	$\{3\}$	

We will only write down (as a 3-dimensional vector, corresponding to players 1, 2 and 3) what load do the three players perceive on the edge they use in each step (exluding themselves):

i. (21, 20, 16)

ii. (21, 20, 7)

iii. (8, 19, 8)

iv. (8, 15, 8)

v. (8, 16, 16)

vi. (20, 16, 16)

The existence of the above infinite improvement cycle forbids the existence of any potential function. □

δ-Homogeneous Stochastic Players. In this case, players have the same confidence level δ, but the participation probability p_i of each player i may be different. We next show how to efficiently compute a PNE in parallel-link networks. Specifically, we describe the p-Decreasing Greedy Best Response algorithm, or pDGBR in short, and show that it always results in a PNE:

- Sort players in *non-increasing* order of their p_i's.
- Insert one player at a time, according to the previous order, to the edge that corresponds to her best response move.
- Repeat until all players are inserted.

Theorem 6.4. The **pDGBR** procedure computes a PNE for Congestion Games with Stochastic δ-Homogeneous Players on parallel-link networks with general latecy functions in time $O(n \cdot m + n^2)$.

Proof. The proof comes by induction on the number of players. The induction hypothesis is that before the next insertion we are at a PNE.

Assume that we are in the middle of the procedure, in a PNE, and player i has just chosen edge e. Players on other edges do not deviate because the only edge changed is e and it got an extra player. What remains to prove is that players on e do not deviate.

Let k be a player on e, inserted in a previous step ($p_k \geq p_i$). In order to show that k doesn't deviate, it suffices to show that after step i, say under configuration s (with $s_i = s_k = e$), we have that $c_k(s) \leq c_i(s)$, as edge e is a best response strategy for player i and the cost that i and k perceive on any other edge is the same.

Consider $c_k(s)$ and $c_i(s)$. We have:

$$c_k(s) = \min\left\{t : \Pr[d_e(1 + X_i + \sum_{j\neq i,k: s_j=e} X_j) \leq t] \geq \delta\right\},$$

$$c_i(s) = \min\left\{t : \Pr[d_e(1 + X_k + \sum_{j\neq i,k: s_j=e} X_j) \leq t] \geq \delta\right\}.$$

Since $p_k \geq p_i$, for any $r \in \mathbb{N}$:

$$\begin{aligned}\Pr[X_k + \sum_{j\neq i,k:\ s_j=e} X_j \leq r] &= \Pr[\sum_{j\neq i,k:\ s_j=e} X_j \leq r] - \Pr[\sum_{j\neq i,k:\ s_j=e} X_j = r] \cdot p_k \\ &\leq \Pr[\sum_{j\neq i,k:\ s_j=e} X_j \leq r] - \Pr[\sum_{j\neq i,k:\ s_j=e} X_j = r] \cdot p_i \\ &= \Pr[X_i + \sum_{j\neq i,k:\ s_j=e} X_j \leq r].\end{aligned}$$

Thus, since d_e's are non decreasing,

$$\Pr[d_e(1 + X_k + \sum_{j\neq i,k:\ s_j=e} X_j) \leq d_e(r+1)] \leq \Pr[d_e(1 + X_i + \sum_{j\neq i,k:\ s_j=e} X_j) \leq d_e(r+1)]$$

and so $c_k(s) \leq c_i(s)$, as needed.

The total time needed for the procedure is $O(n \cdot m + n^2)$, as at each step i, the computations for the newly inserted player take $O(m + i^2)$ time and we can use memoization to avoid recalculations. □

We can also show that games in this class admit a potential function, which can be thought of as a generalized lexicographic potential function in two dimensions.

Theorem 6.5. Congestion Games with Stochastic δ-Homogeneous Players on parallel-link networks are potential games.

Proof. We will define a two dimensional lexicographic potential function. For that, we define, for each edge e and configuration s, a two dimensional vector $v_{e,s}$ and a total order on these vectors. Right after, for s, we define a vector w_s that as components has exactly the vectors $\{v_{e,s}\}_{e\in E}$ ordered increasingly (with the usual vector ordering). Any improvement step that turned the configuration from s to s' is such that $w_s < w_{s'}$. Any strictly decreasing function on vectors w_s works as (generalized) potential. Technical details follow.

Let $c_e(s) = \min\left\{t : \Pr[d_e(1 + N_e(s)) \leq t] \geq \delta\right\}$ be the *outside δ-cost* of edge e under s, i.e. the cost that any player not in e computes for e when she considers moving to e. We have

$$c_e(s) = c_i(s_{-i}, e),\ \forall i : s_i \neq e \tag{6.1}$$

$$c_e(s) \geq c_i(s),\ \forall i : s_i = e. \tag{6.2}$$

Let $v_{e,s} = (c_e(s), s_e)$ and consider the standard ordering on these pairs:

- $(x_1, y_1) < (x_2, y_2)$ iff $x_1 < x_2$ or ($x_1 = x_2$ and $y_1 < y_2$).
- $(x_1, y_1) = (x_2, y_2)$ iff $x_1 = x_2$ and $y_1 = y_2$.
- $(x_1, y_1) > (x_2, y_2)$ otherwise.

For a configuration s, let w_s be the vector that consists of the pairs $\{v_{e,s}\}_{e\in E}$ in increasing order. We are going to show that after an improving step, the new configuration s' is such that $w_s < w_{s'}$.

Assume that player i did an improvement step by moving from e to e' and let s be the configuration with $s_i = e$ and s' the configuration with $s'_i = e'$. Player i deviated because $c_i(s) > c_i(s')$. It is $c_i(s') = c_i(s_{-i}, e') = c_{e'}(s)$ and, by (6.2), $c_e(s) \geq c_i(s)$. So, we get $c_e(s) > c_{e'}(s)$, which implies that

$$v_{e',s} < v_{e,s}.$$

Thus, if we consider the coordinates of w_s, we know that $v_{e,s}$ is after $v_{e',s}$ and so in order to show $w_s < w_{s'}$, it suffices to show that $v_{e',s} < v_{e',s'}$ (inequality 6.3) and $v_{e',s} < v_{e,s'}$ (inequality 6.4).

By the improving step, the only pairs that changed are $v_{e,s}$ and $v_{e',s}$. In fact, it is $s_e > s'_e$ and $s_{e'} < s'_{e'}$ and it might be $c_e(s) > c_e(s')$ or $c_{e'}(s) < c_{e'}(s')$. Whichever is the case, we have:

$$v_{e',s} < v_{e',s'} \tag{6.3}$$

Because i did an improving step to e' and by $s_i \neq e'$ and (6.1) we get $c_i(s) > c_i(s_{-i}, e') = c_{e'}(s)$. Also, by $s'_i \neq e$ and (6.1) we get $c_i(s) = c_i(s'_{-i}, e) = c_e(s')$. Thus, $c_{e'}(s) < c_e(s')$ and

$$v_{e',s} < v_{e,s'} \tag{6.4}$$

Let Φ be any strictly decreasing function on w_s. Clearly, for any configuration s, any edges e, e' and any player i it is

$$c_i(s_{-i}, e) - c_i(s_{-i}, e') > 0 \Rightarrow \Phi(s_{-i}, e) - \Phi(s_{-i}, e') > 0$$

and thus Φ is a generalized potential. □

6.2.3 Price of Anarchy for Games with Affine Latencies

In Stochastic Congestion Games with Stochastic Players, the *social cost* of a configuration s is defined as $C(s) = \mathbf{E}\left[\sum_{i\in N} X_i D_i(s)\right]$, as this seems to be the most natural generalization of the definition of the Social Cost in standard atomic CGs.

Letting o denote an optimal configuration, i.e. $c(o) = \min_s\{C(s)\}$, the Price of Anarchy is formally defined as $PoA = \max\left\{\frac{C(s)}{C(o)} : \ s \text{ is a PNE}\right\}$.

In the following we convert $C(s)$ to a more convenient form and right after we give upper and lower bounds on the PoA for games with affine latency functions as the expected values related to the random distributions $d_e(N_e(s))$ for general latency functions are hard to handle.

As already noted, $D_i(s) = D_{s_i}(s)$ with $X_i = 1$, and so $X_i \cdot D_i(s) = X_i \cdot D_{s_i}(s)$. Thus,

$$C(s) = \mathbf{E}\left[\sum_{i\in N} X_i \cdot D_{s_i}(s)\right] = \sum_e \mathbf{E}[N_e(s) \cdot d_e(N_e(s))].$$

For affine latency functions of the form $d_e(k) = a_e k + b_e$, it is

$$C(s) = \sum_e \mathbf{E}[N_e(s)\cdot(a_e N_e(s)+b_e)] = \sum_e \left[a_e(\mathbf{E}[N_e(s)]^2+\mathbf{Var}[N_e(s)])+b_e\mathbf{E}[N_e(s)]\right].$$

Theorem 6.6. Congestion Games with Stochastic Players and affine latency functions on parallel-link networks have $PoA = O(n)$.

Proof. Let $d_e(k) = a_e k + b_e$ denote the latency of an edge e. Two things should be pointed out for the proof.

i) The costs that players perceive on each edge is almost at least as large as it would be if they were considering as load the expectation of other players on each edge plus themselves, i.e. $c_i(s) \geq a_e \mathbf{E}[\sum_{j:\ e=s_j} X_j] - a_e + b_e$, for any s and $i : s_i = e$.

This is the case as, if we assume that edges have affine delay functions, $d_e(k) = a_e k + b_e$, then the cost that player i perceives for edge e under a configuration s with $s_i = e$ is

$$\begin{aligned} c_i(s) &= \min\Big\{t \,|\, \Pr\Big[d_e\Big(1 + \sum_{j\neq i:\ e=s_j} X_j\Big) \leq t\Big] \geq \delta_i\Big\} \\ &= \min\Big\{t \,|\, \Pr\Big[a_e\Big(1 + \sum_{j\neq i:\ e=s_j} X_j\Big) + b_e \leq t\Big] \geq \delta_i\Big\} \\ &= a_e\Big(1 + \min\Big\{t \,|\, \Pr[\sum_{j\neq i:\ e=s_j} X_j \leq t] \geq \delta_i\Big\}\Big) + b_e. \end{aligned}$$

For the median and the mean of the random variable $\sum_{j\neq i:\ e=s_j} X_j$ it is (see e.g [80])

$$\Big|\mathbf{E}[\sum_{j\neq i:\ e=s_j} X_j] - \min\Big\{t \,|\, \Pr[\sum_{j\neq i:\ e=s_j} X_j \leq t\Big] \geq 1/2\Big\}\Big| \leq 1.$$

Thus, for player $i : s_i = e$,

$$\begin{aligned} c_i(s) &\geq a_e\Big(1 + \min\Big\{t \,|\, \Pr[\sum_{j\neq i:\ e=s_j} X_j \leq t] \geq 1/2\Big\}\Big) + b_e \\ &\geq a_e\mathbf{E}[\sum_{j\neq i:\ e=s_j} X_j] + b_e \geq a_e\mathbf{E}[\sum_{j:\ e=s_j} X_j] - a_e + b_e \end{aligned}$$

ii) At equilibrium, all players on the used edges perceive a cost not greater than $n(a + b)$, where $a + b = \min_e\{a_e + b_e\}$.

This is the case, or else a player that perceives a greater cost would have an incentive to deviate to the edge e with $d_e(k) = ak + b$.

To prove the theorem, let f be a Nash equilibrium and o be an optimal configuration. We wish to bound the cost of f, $C(f)$, by a factor of the form $n \cdot C(o)$.

Denote $F_e \equiv N_e(f)$ and $O_e \equiv N_e(o)$. We have

$$\begin{aligned} C(f) &= \sum_e \Big(a_e(\mathbf{E}[F_e]^2 + \mathbf{Var}[F_e]) + b_e\mathbf{E}[F_e]\Big) = \sum_e \mathbf{E}[F_e]\Big(a_e\mathbf{E}[F_e] + b_e + a_e\frac{\mathbf{Var}[F_e]}{\mathbf{E}[F_e]}\Big) \\ &\leq \sum_e \mathbf{E}[F_e]\Big(c_{max} + a_e + a_e\frac{\mathbf{Var}[F_e]}{\mathbf{E}[F_e]}\Big) \leq 3\sum_e \mathbf{E}[F_e]c_{max}. \end{aligned}$$

where c_{max} is the greatest cost that a player perceives under f. The inequalities follow from point (i) above, $c_{max} \geq a_e$ for all e under use and $\frac{\mathbf{Var}[F_e]}{\mathbf{E}[F_e]} \leq 1$.

From point (ii) above, with $a + b = \min_e\{a_e + b_e\}$, we get that

$$\begin{aligned} C(f) &\leq 3c_{max} \sum_e \mathbf{E}[F_e] \leq 3n(a+b) \sum_e \mathbf{E}[F_e] = 3n(a+b) \sum_{i \in N} p_i \\ &= 3n(a+b) \sum_e \mathbf{E}[O_e] \leq 3n \sum_e \mathbf{E}[O_e](a_e + b_e) \\ &\leq 3n \sum_e \mathbf{E}[O_e]\left(a_e \frac{\mathbf{E}[O_e]^2 + \mathbf{Var}[O_e]}{\mathbf{E}[O_e]} + b_e\right) = 3nC(o). \end{aligned}$$

The last inequality follows by $\frac{\mathbf{E}[O_e]^2 + \mathbf{Var}[O_e]}{\mathbf{E}[O_e]} = \frac{\mathbf{E}[O_e^2]}{\mathbf{E}[O_e]} \geq 1$. □

Next, we give a class of fully symmetric games with stochastic players, on singleton strategies and linear latency functions, that has $PoA = \Omega(n)$.

Theorem 6.7. There are Congestion Games with Homogeneous Players that have $PoA = \Omega(n)$.

Proof. Let $k \in \mathbb{N}$. Consider the parallel links game with n players and $k + 1$ edges. The first edge, e_1, has delay function $d_1(x) = x$ and the others, $e_2, \ldots, e_{k+1}$, have $d_j(x) = (n - k)x$, $j = 2, \ldots, k + 1$. Players play with (common) probability p and are fearful, i.e. all of them have $\delta = 1$, and so they consider that all players on their edge use it.

Let f be the configuration where $n - k$ players are on e_1 and the other k players are one on each of the k remaining edges. This is a PNE, since each player perceives a cost of $n - k$.

All edges $\{e_i\}_{i=1\ldots k+1}$ have $b_{e_i} = 0$. So, the social cost of f is

$$C(f) = \sum_e \left[a_e(\mathbf{E}[N_e(f)]^2 + \mathbf{Var}[N_e(s)])\right]$$

For e_1 it is: $a_{e_1}(\mathbf{E}[N_{e_1}(f)]^2 + \mathbf{Var}[N_{e_1}(s)]) = (n - k)^2p^2 + (n - k)p(1 - p)$
For all $e \in \{e_i\}_{i=2\ldots k+1}$ it is: $a_e(\mathbf{E}[N_e(f)]^2 + \mathbf{Var}[N_e(s)]) = p(n - k)$.
So

$$C(f) = (n - k)^2p^2 + (n - k)p(1 - p) + kp(n - k).$$

Now let configuration o be the one in which every player plays e_1. It is

$$C(o) = n^2p^2 + np(1 - p)$$

So, for $k = \frac{n}{2}$ and $p = \frac{1}{n}$

$$PoA \geq \frac{C(f)}{C(o)} = \frac{(n - k)^2p^2 + (n - k)p(1 - p) + kp(n - k)}{n^2p^2 + np(1 - p)} = \frac{1}{2} + \frac{n - 1}{8 - \frac{4}{n}} = \Omega(n),$$

as needed. □

6.3 Congestion Games with Stochastic Edges

6.3.1 The Model

In Congestion Games with Stochastic Edges, players are deterministic, i.e. they always participate in the game and are represented only by their confidence levels δ_i. On the other hand, edges have a stochastic behavior. For an edge e, its latency function is an independent random variable:

$$d_e(k) = \begin{cases} f_e(k), & \text{with probability } 1 - p_e \\ g_e(k), & \text{with probability } p_e. \end{cases}$$

The cost of an edge e with load k is: $X_e(k) = (1-p_e)\cdot f_e(k) + p_e \cdot g_e(k)$ and the cost of a player i with strategy s_i, is the random variable: $D_i(s) = \sum_{e \in s_i} X_e(s_e)$

The cost that player i perceives is: $c_i(s) = \min\left\{t \mid \Pr\left[D_i(s) \leq t\right] \geq \delta_i\right\}$ and the *social cost* of a configuration s is defined as: $C(s) = \mathbf{E}\left[\sum_{i \in N} D_i(s)\right]$, i.e. similar to the stochastic players case.

Classes of congestion games. Here we only define two classes of congestion games with stochastic edges:

- *homogeneous*, where all players have the same δ_i's.
- *heterogeneous*, where players may have different δ_i's.

The class of congestion games with stochastic edges and homogeneous players boils down to the class of potential games, while the class of congestion games with stochastic edges and heterogeneous players boils down to the class of congestion games with *player specific cost functions* ([63]).

6.3.2 Stochastic Edges on Parallel Links: Existence and Computation of PNE

In the following, we restrict ourselves to Congestion Games with Stochastic Edges on parallel-link networks, and investigate the existence and the efficient computation of PNE for the two cases considered above.

Homogeneous Stochastic Players. If the players are homogeneous, stochastic congestion games on parallel-links are equivalent to standard congestion games on parallel-links (but with possibly different latencies), because the (risk-averse) individual cost of each player in a configuration s depends only on the link e and its congestion s_e.

Theorem 6.8. Stochastic Congestion Games with Stochastic Edges and Homogeneous Players on parallel-link networks are potential games. Moreover, a PNE can be computed in time $O(n \cdot m)$.

Proof. Recall that $D_i(s) = \sum_{e \in s_i} X_e(s_e)$ and the cost that players perceive is $c_i(s) = \min\left\{t \mid \Pr[D_i(s) \leq t] \geq \delta_i\right\}$ and observe that since players have the same δ, the cost that any player i perceives on edge e when there are s_e players in total on edge e is the same for all i, and equal to:

$$c_e(s_e) = \begin{cases} f_e(s_e), & \text{if } 1 - p_e \geq \delta \\ g_e(s_e), & \text{otherwise} \end{cases}$$

We now define the standard potential function:

$$\Phi(s) = \sum_e \sum_{j=1}^{s_e} c_e(s)$$

It is easy now to verify that the above function is indeed an exact potential function. Observe that there is no restriction for $p_e, f_e(n)$ and $g_e(n)$.

Regarding the computation of a PNE, we can again use a standard *Greedy Best Response* algorithm. We insert the players in the game, one by one, and the player just inserted does a best response move. Since all players perceive the same costs, it is clear that at each step we have a PNE, and so, when the procedure terminates, we are still in a PNE. As for the time needed, at each step the calculation of the best response move takes time $O(m)$, and so the total time needed is $O(n \cdot m)$. □

Heterogeneous Stochastic Players. In this case, a stochastic game is equivalent to a congestion game on parallel links with player-specific payoffs [63], as the (risk-averse) individual cost of each player in a configuration s depends only on the link e, its congestion s_e, and i's confidence level δ_i. Thus, we obtain:

Corollary 6.9. Congestion Games with Heterogenous Stochastic Players and Stochastic Edges on parallel-link networks admit a PNE. Moreover, a PNE can be computed in polynomial time.

In ([63]) it is proved that congestion games with player specific payoff functions always possess a PNE. We now prove that the class of Congestion Games with Stochastic Edges does not admit any kind of potential, even if we restrict ourselves to affine latency functions.

Theorem 6.10. There is no potential function for the class of Congestion Games with Stochastic Edges and deterministic risk-averse players.

Proof. We consider 3 players and 3 links, with the same "failure" probabilities. The latency functions are:

$$f_1(k) = \begin{cases} 3k+21, & \text{with probability } 0.8 \\ 5k+22, & \text{with probability } 0.2 \end{cases}$$

$$f_2(k) = \begin{cases} 6k+16, & \text{with probability } 0.8 \\ 22k+5, & \text{with probability } 0.2 \end{cases}$$

$$f_3(k) = \begin{cases} k+24, & \text{with probability } 0.8 \\ 25k+1, & \text{with probability } 0.2 \end{cases}$$

Players 1 and 2 have confidence level $\delta_1 = \delta_2 = 0.7$ and player 3 has confidence level $\delta_3 = 0.9$. We now construct the following cycle:

e_1	e_2	e_3
$\{1,2\}$	$\{3\}$	
$\{1,2\}$		$\{3\}$
$\{2\}$		$\{1,3\}$
	$\{2\}$	$\{1,3\}$
	$\{2,3\}$	$\{1\}$
$\{1\}$	$\{2,3\}$	
$\{1,2\}$	$\{3\}$	

One can verify that this is a better response cycle and thus no potential function exists. □

6.3.3 Price of Anarchy

The risk aversion of the players combined with their selfish behavior may give unlimited degradation to the network.

Theorem 6.11. There are Stochastic Congestion Games with Stochastic Edges that have unbounded PoA.

Proof. We will use a simple network with two parallel edges and two players:

$$f_1(k) = \begin{cases} ak+1, & \text{with probability } 0.50+\epsilon \\ Ak+1, & \text{with probability } 0.50-\epsilon, \end{cases}$$

$$f_2(k) = 2ak+1+\epsilon, \text{with probability } 1.$$

(Assume that $\epsilon < 1/2$.)

Both players have confidence level $\delta_1 = \delta_2 = \delta = 0.5$. We assume that $A > 2a$. As a result, both players "see" the good part of edge e_1, and so

they both prefer it. Let $s = (e_1, e_1)$ be the configuration vector of the PNE. The social cost of s is $C(s) = 2[(2a + 1)(0.5 + \epsilon) + (2A + 1)(0.5 - \epsilon)]$, while an optimal configuration is $o = (e_2, e_2)$ with social cost $C(o) = 2(4a + 1 + \epsilon)$. We have:

$$\begin{aligned} PoA = \frac{C(s)}{C(o)} &= \frac{2[(2a + 1)(0.5 + \epsilon) + (2A + 1)(0.5 - \epsilon)]}{2(4a + 1 + \epsilon)} \\ &= \frac{(2a + 1)(0.5 + \epsilon)}{4a + 1 + \epsilon} + \frac{(2A + 1)(0.5 - \epsilon)}{4a + 1 + \epsilon} \\ &\geq \frac{(0.5 - \epsilon)(2A + 1)}{4a + 1 + \epsilon} \\ &\geq \frac{(0.5 - \epsilon)2A}{4a + 1 + \epsilon} \\ &\geq \frac{(0.5 - \epsilon)A}{2a + 1} \end{aligned}$$

Thus, $PoA \to \infty$, if $\frac{(0.5-\epsilon)A}{2a+1} \to \infty$ (e.g. $\epsilon < 0.4$ and $\frac{A}{2a+1} \to \infty$).

□

Chapter 7

Improving Selfish Routing through Risk Aversion

In this chapter, we investigate how and to which extent one can exploit risk-aversion and modify the perceived latencies of the players so that the Price of Anarchy (PoA) wrt. the total latency of the players is improved. The starting point is to introduce some small (and carefully selected) random perturbations to the edge latencies so that the expected latency does not change, but the perceived cost of the players increases, due to risk-aversion. To provide a simple and general theoretical model of this behavior, we introduce γ-modifiable routing games where the latency function of each edge e can increase from $\ell_e(x)$ to $(1+\gamma_e)\ell_e(x)$, for some selected $\gamma_e \in [0,\gamma]$. For γ-modifiable games in parallel-links and in series-parallel networks, we fully characterize the values of γ for which γ-bounded latency modifications can decrease the PoA to 1. Moreover, we show how to (efficiently) compute a set of γ-bounded latency modifications so that the PoA of the resulting game improves significantly as γ increases. E.g., for linear latencies, the resulting PoA is at most $\max\{1,(1-(1-\gamma)^2/4)^{-1}\}$. We prove that our PoA analysis is tight, even for two parallel links, and also discuss the difficulty of extending our characterization and our construction to general networks.

7.1 Introducing γ-modifiable CGs

In order to introduce γ-modifiable CGs, we first discuss how (typically small) random perturbations in the edge latencies can be performed so that the expected latency does not change, but the latency perceived by the players increases, due to risk aversion. E.g., let us consider an edge e

with latency function $d_e(x)$ where we can increase the latency temporarily up to $(1 + a_1)d_e(x)$ and decrease it temporarily (and for relatively short time intervals) up to $(1 - a_2)d_e(x)$. If we implement the former change with probability p_1 and the latter with probability $p_2 < 1 - p_1$ (the probabilities here essentially correspond to proportions of time in which e operates in each state), the latency function of e in a given time step is a random variable $\ell_e(x)$ with expectation:

$$\mathbf{E}[\ell_e(x)] = [p_1(1 + a_1) + p_2(1 - a_2) + (1 - p_1 - p_2)]d_e(x)$$

Adjusting p_1 and p_2 (and possibly a_1 and a_2) so that $p_1 a_1 = p_2 a_2$, we have $\mathbf{E}[\ell_e(x)] = d_e(x)$, i.e., for any given flow, the expected delay through e does not change. On the other hand, if the players are risk-averse and their individual cost is given by an $(1 - p_1 + \varepsilon)$-quantile of the delay distribution (e.g., as in [68, 6]), for some $\varepsilon > 0$, the latency perceived by the players on e is $(1 + a_1)d_e(x)$. Similarly, if the individual cost of the risk-averse players are given by the expectation plus the standard deviation of the delay distribution (e.g., as in [67]), the latency perceived by the players on e is $(1 + \sqrt{p_1 a_1^2 + p_2 a_2^2})d_e(x)$. In both cases, we can have a significant increase in the latency perceived by the risk-averse players on e, while the expected latency remains unchanged. A similar result could be achieved with any latency distribution on e (possibly more sophisticated and with larger support), as long as its expectation is $d_e(x)$.

In most practical situations, the increase and, especially, the decrease in the latency functions that can be implemented are bounded (and relatively small). The same is particularly true for the proportion of time in which an edge can operate in an "abnormal" state of increased or decreased latency. Combined with the formula providing the individual cost of the risk-averse players, these factors determine an upper bound γ_e on the multiplicative increase of the latency perceived by the players on each edge e. Thus, motivated by such considerations, we introduce the so-called *γ-modifiable selfish routing games* as a simple and general abstraction of how one can exploit risk-aversion towards improving the PoA of selfish routing.

γ-Modifiable Routing Games. A selfish routing game $\mathcal{G} = (G, d, r)$ is *γ-modifiable* if for each edge $e \in G$, we can choose a $\gamma_e \in [0, \gamma]$ and change the edge latency functions perceived by the players from $d_e(x)$ to $(1+\gamma_e)d_e(x)$ by small random perturbations, as discussed above. Any vector $\Gamma = (\gamma_e)_{e \in E}$, where $\gamma_e \in [0, \gamma]$ for each edge e, is called a γ-modification of $\mathcal{G}$. Given a γ-modification Γ, we let $\mathcal{G}^\Gamma$ denote the γ-modified routing game $\mathcal{G}^\Gamma = (G, (1+\Gamma)d, r)$, with a latency function $(1+\gamma_e)d_e(x)$ on each edge e, obtained from $\mathcal{G}$. To simplify notation, we sometimes write G^Γ, instead of $\mathcal{G}^\Gamma$.

Given a flow f, the latency perceived by the players on a path p in $\mathcal{G}^\Gamma$ is calculated wrt the modified latencies and is equal to $d_p^\Gamma(f) = \sum_{e\in p}(1+\gamma_e)d_e(f_e)$ [1]. A flow f is a *Nash flow* for the modified game, if it routes all traffic on minimum perceived latency paths, i.e., if for every path p with $f_p > 0$, and every path p', $d_p^\Gamma(f) \leq d_{p'}^\Gamma(f)$. Again, to simplify notation, we usually write $d_p(f)$, instead of $d_p^\Gamma(f)$, as long as it is clear from the context that the path latencies are wrt. the modified game $\mathcal{G}^\Gamma$.

Given a routing game $\mathcal{G}$, we say that a flow f is *γ-enforceable*, or simply *enforceable*, if there exists a γ-modification Γ of $\mathcal{G}$ such that f is a Nash flow of $\mathcal{G}^\Gamma$.

We always assume that γ-modifications keep the expected latency functions unchanged. Since for any flow f, the expected latency of any edge e in $\mathcal{G}^\Gamma$ is equal to the latency of e in $\mathcal{G}$ under f, the (expected) total latency of f in both $\mathcal{G}^\Gamma$ and $\mathcal{G}$ is equal to $C(f) = \sum_{e\in E} f_e d_e(f_e)$. Hence, the optimal flow o of $\mathcal{G}$ is also an optimal flow of $\mathcal{G}^\Gamma$.

The Price of Anarchy $\text{PoA}(\mathcal{G}^\Gamma)$ of the modified game $\mathcal{G}^\Gamma$ is equal to $C(f)/C(o)$, where f is the Nash flow of $\mathcal{G}^\Gamma$. For a γ-modifiable game $\mathcal{G}$, the Price of Anarchy of $\mathcal{G}$ under γ-modifications, denoted $\text{PoA}_\gamma(\mathcal{G})$, is the best PoA that we can achieve by some γ-modification. Formally, $\text{PoA}_\gamma(\mathcal{G}) = \min\{PoA(G^\Gamma) | \Gamma \text{ is a } \gamma\text{-modification of } \mathcal{G}\}$. For routing games with cost functions in a class $\mathcal{D}$, $\text{PoA}_\gamma(\mathcal{D})$ denotes the maximum $\text{PoA}_\gamma(\mathcal{G})$ over all γ-modifiable games $\mathcal{G}$ with latency functions in class $\mathcal{D}$.

Connection to Marginal-Cost Tolls. A sufficient condition for the optimal flow to be γ-enforceable can be obtained through optimal marginal-cost tolls, that assign an additive toll of $o_e d'_e(o_e)$ to any edge e (see e.g., [77]).

Proposition 8. *Let o be the optimal flow of a γ-modifiable instance $\mathcal{G}$. If for all links e with $o_e > 0$, $\frac{o_e d'_e(o_e)}{d_e(o_e)} \leq \gamma$, then o is γ-enforceable in $\mathcal{G}$.*

Proof. For each edge e with $o_e > 0$ (and thus, with $d_e(o_e) > 0$), we let $\gamma_e = \frac{o_e d'_e(o_e)}{d_e(o_e)}$, while for each edge e with $o_e = 0$, we let $\gamma_e = 0$. By hypothesis, this defines a γ-modification Γ of $\mathcal{G}$. Moreover, if we change the latency functions of $\mathcal{G}$ from $d_e(x)$ to $d_e(x) + x d'_e(x) = (1+\gamma_e)d_e(x)$, for all edges e, [77, Cor. 2.4.6] implies that the optimal flow o of $\mathcal{G}$ is a Nash flow of the modified game $\mathcal{G}^\Gamma$. □ □

[1] To simplify the model and make it easily applicable to general networks, we make the convenient assumption that the latency modifications (and the resulting individual costs of the players) are separable, although most common notions of individual cost for risk-averse players result in non-separable costs (see e.g., [72, 68, 67, 6], but see also [71], where the simplifying assumption of independence among randomized schedulers of different edges also implies the separability of individual costs). The separable costs assumption only affects the extension of our results to series-parallel networks.

Proposition 8 demonstrates that our approach is applicable to any routing game. Moreover, if $\mathcal{G}$ has polynomial latency functions of degree d, then the optimal flow is γ-enforceable for any $\gamma \geq d$. However, Proposition 8 only provides a necessary condition and does not fully characterize the class of γ-modifiable routing games for which the optimal flow is enforceable (see also Section 7.2). Thus, we develop, in the next sections, a complete characterization of γ-modifiable routing games in parallel-link and in series-parallel networks with γ-enforceable optimal flows.

7.2 Modifying Routing Games in Parallel-Link Networks

We proceed to study γ-modifiable instances in parallel-link networks. We first deal with the question of characterizing the γ-modifiable instances where we can enforce the optimal flow. Then, we provide matching upper and lower bounds on the PoA_γ for γ-modifiable instances where the optimal flow is not enforceable. We also show how to compute γ-modifications that guarantee these PoA_γ bounds.

We note that the converse of Proposition 8 is not necessarily true. E.g., let us consider a $\frac{1}{2}$-modifiable instance $\mathcal{G}$ on 2 parallel links with $d_1(x) = x$, $d_2(x) = x + 2$ and $r = 3$. The optimal flow is $o = (2, 1)$, and thus, enforceable in $\mathcal{G}$. On the other hand, the ratios of the marginal cost tolls to the optimal latencies are 1 and $1/4$, for links 1 and 2, respectively. Hence, Proposition 8 only ensures that the optimal flow is enforceable if $\mathcal{G}$ is 1-modifiable. Therefore, we proceed to develop a characterization of γ-modifiable games for which the optimal flow is enforceable.

Theorem 7.1. Let $\mathcal{G}$ be a γ-modifiable game on parallel links and let o be the optimal flow of $\mathcal{G}$. The following are equivalent:

(i) o is γ-enforceable in $\mathcal{G}$.

(ii) for all links $e, e' \in E$ with $o_e > 0$, $d_e(o_e) \leq (1 + \gamma)d_{e'}(o_{e'})$.

Proof. $(i) \rightarrow (ii)$: Let $\Gamma = (\gamma_e)_{e \in E}$ be a γ-modification that makes o the Nash flow of $\mathcal{G}^\Gamma$ and consider a pair of edges e and e'. Assume w.l.o.g. that $d_{e'}(o_{e'}) \leq d_e(o_e)$ and $o_e > 0$. Flow o is a Nash flow in the modified network and thus $(1 + \gamma_e)d_e(o_e) \leq (1 + \gamma_{e'})d_{e'}(o_{e'})$ which directly implies

$$d_e(o_e) \leq (1 + \gamma_e)d_e(o_e) \leq (1 + \gamma_{e'})d_{e'}(o_{e'}) \leq (1 + \gamma)d_{e'}(o_{e'})$$

$(ii) \rightarrow (i)$: Assume that for any pair of edges e and e', if $o_e > 0$ then $d_e(o_e) \leq (1 + \gamma)d_{e'}(o_{e'})$. Let $d_{max} = \max_e\{d_e(o_e) | o_e > 0\}$ be the maximum cost between all used edges under o. Assign for each edge e with $o_e > 0$,

$\gamma_e = \frac{d_{max} - l_e(o_e)}{d_e(o_e)}$. It is $\gamma_e \leq \gamma$, for all e, as by hypothesis $d_{max} \leq (1+\gamma)d_e(o_e)$, for all e. Assign for each edge e with $o_e = 0$, $\gamma_e = 0$. Let $\Gamma = (\gamma_e)_{e \in E}$. It is $\forall e : o_e > 0 \Rightarrow (1+\gamma_e)d_e(o_e) = d_{max}$. Moreover, $\forall e' : o_{e'} = 0 \Rightarrow d_{e'}(o_{e'}) \geq d_{max}$ as by (e.g.) corollary 2.4.6 in [77], for an $e' : o_{e'} = 0$ and an $e : o_e > 0$ and $d_e(o_e) = d_{max}$, it is $d_{e'}(o_{e'}) \geq d_e(o_e) + o_e d'_e(o_e) \geq d_{max}$. Thus o is a Nash flow in $\mathcal{G}^\Gamma$. □

Next, in the following Lemma, for any γ-modifiable instance $\mathcal{G}$ with optimal solution o, we prove the existence of a pair (f,Γ) with specific suitable properties. This Lemma is the key both for proving the bound on the PoA_γ and guaranteing an efficient computation of a pair (f,Γ) that falls in that bound.

Lemma 7.2. Let $\mathcal{G} = (G, d, r)$ be a parallel-links γ-modifiable instance and let o be the optimal flow of $\mathcal{G}$. There is a feasible flow f and a γ-modification Γ of $\mathcal{G}$ such that

i) f is a Nash flow in G^Γ.

ii) for any edge e: if $f_e < o_e$ then $\gamma_e = 0$ and if $f_e > o_e$ then $\gamma_e = \gamma$.

Proof. Consider $\mathcal{G} = (G, d, r)$ and its optimal solution o. If o can be γ-enforced, then by definition, there is a γ-modification Γ for which o is the Nash flow of $\mathcal{G}^\Gamma$. Thus for $f = o$ and the above Γ, the Lemma holds. For all other cases we will use induction on the number of edges in G.

In the base case of a single edge, under any rate r, f and o coincide and thus under any modification, the Lemma holds.

For the inductive step let e_m be a used edge with maximum cost under o. Remove e_m from the network, remove the flow through e_m, i.e. o_{e_m}, and let the instance $\mathcal{G}_m = (G_m, d, r' = r - o_e)$ denote the new instance. For $\mathcal{G}_m$, by induction hypothesis, the Lemma holds. Thus there is a flow f' and a γ-modification $\Gamma' = (\gamma'_e)_{e \in E_m}$ of $\mathcal{G}_m$ such that: *i)* f' is the Nash flow in $G_m^{\Gamma'}$, *ii)* for any edge e: if $f'_e < o'_e$ then $\gamma'_e = 0$ and if $f'_e > o'_e$ then $\gamma'_e = \gamma$, where o' is the optimal solution of $\mathcal{G}_m$.

Now we will put back edge e_m and the removed flow. If there is a γ_{e_m} so as $(1+\gamma_{e_m})d_{e_m}(o_{e_m}) = L(G_m, (1+\Gamma')d, r')$ then we can add γ_{e_m} to Γ' and get Γ and the Lemma will hold for $\mathcal{G}^\Gamma$, with f being such that $f_{e_m} = o_{e_m}$ and $f_e = f'_e$ for all other edges. Else it should be $d_{e_m}(o_{e_m}) > L(G_m, (1+\Gamma')d, r')$ as for any used edge e under f', with $f'_e < o_e$, it is $d_e(f'_e) \leq d_e(o_e) \leq d_{e_m}(o_{e_m})$ and for any used edge e' with $f'_{e'} \geq o_{e'}$ it is $d_e(f'_e) = (1+\gamma'_e)d_{e'}(f'_{e'}) \geq d_{e'}(f'_{e'})$ by properties *i)* and *ii)* of the induction hypothesis.

It remains to handle the case $d_{e_m}(o_{e_m}) > L(G_m, (1+\Gamma')d, r')$. Intuitively we will reroute flow from e_m to the rest of the edges so as the equilibrium property in G_m is not destroyed. Our final goal is to make e_m cost equal

to the Nash flow cost of G_m. During this procedure though, we have to be careful also not to destroy property *ii)*. This could happen if we carelessly push flow to edges of G_m with $f'_e < o_e$. To take care of this, while rerouting, if such an edge e gets flow equal to o_e we stop rerouting through it but, in order to keep the equilibrium property, we unlock its γ_e and change its value[2] until it becomes γ. It also may be that during this procedure e_m gets empty of flow but this case is covered by the induction hypothesis for the instance (G_m, r). Details follow.

Let $0 \leq x_m \leq o_{e_m}$ be the maximum value (amount of flow) such that:

(a) there exist a γ-modification Γ^* of G_m such that under flow $r' + x$, the Nash flow flow f^* of $G_m^{\Gamma^*}$ is such that for any edge $e \in G_m^{\Gamma^*}$: if $f'_e \geq o_e$ and $\gamma'_e = \gamma$ then $f^*_e \geq f'_e > o_e$ and $\gamma^*_e = \gamma(= \gamma'_e)$, if $f'_e < o_e$ then $f'_e \leq f^*_e \leq o_e$ and $\gamma^*_e = 0(= \gamma_e)$, if $f'_e = o_e$ and $\gamma'_e < \gamma$ then $f^*_e = o_e(= f'_e)$ and $\gamma'_e \leq \gamma^*_e \leq \gamma$.

(b) $d_{e_m}(o_{e_m} - x_m) \geq L(G_m, (1 + \Gamma^*)d, r' + x_m)$

It is $x_m > 0$. To see this, let E_{lock} be the set of edges that have $f'_e = o_e$ and $\gamma'_e < \gamma$ and E_{free} the rest of the used edges under f'. For ϵ small enough, because of continuity, we can reroute flow from e_m to edges in E_{free} so as the equilibrium property in E_{free} is kept, the inequalities of (a) hold and these edges remain in E_{free}, with the inequality of their new flow with o_e having the same direction. By continuity and for ϵ small enough, we can also grow (until the value γ) the γ_e's of the edges in E_{lock} so as the equilibrium property holds in E_{lock}. Combining these facts we can reroute an amount of flow ϵ from e_m to edges in E_{free} and allowably change some of the γ_e values of edges in E_{lock} so as the equilibrium property in G_m is kept with inequality (b) still holding (recall that (b) was not tight).

By the same reasoning, unless (b) is tight or $x_m = o_{e_m}$, it should be that: (I) for x_m and its corresponding Γ^* and f^* there exists an e such that $[(f'_e < o_e$ and $f^*_e = o_e$ and $\gamma^*_e = 0(= \gamma_e))$ or $(\gamma'_e < \gamma$ and $f^*_e = o_e(= f'_e)$ and $\gamma^*_e = \gamma)]$. This means that either an edge in E_{free} under f' with $f'_e < o_e$ will get in E_{lock} under f^* by gaining flow or an edge in E_{lock} under f' will move to E_{free} under f^* by getting $\gamma^*_e = \gamma$ and having $f^*_e = o_e$. Note that an edge $e \in E_{free}$ with $f'_e \geq o_e$ and $\gamma^*_e = \gamma$ cannot get in E_{lock} under any f^* that satisfies (a).

If inequality (b) is tight then we first let $\gamma_{e_m} = 0$ and add it to Γ^* to make a γ-modification Γ. We then let $f_{e_m} = o_{e_m} - x_m$ and $f_e = f^*_e$ for all other edges. The pair (f,Γ) satisfies the Lemma. If inequality (b) is not tight but $x_m = o_{e_m}$[3], then we first let $\gamma_{e_m} = 0$ and add it to Γ^* to make a γ-modification

[2]this doesn't destroy property *ii)*, as the flow through e would be o_e

[3]which implies $d_{e_m}(0) > L(G_m, (1 + \Gamma^*)d, r)$

Γ and then we let $f_{e_m} = 0$ and $f_e = f_e^*$. The pair (f,Γ) satisfies the Lemma.[4]

If inequality (b) is not tight and $x_m < o_{e_m}$ then set $f' = f^*$, $o_{e_m} = o_{e_m} - x_m$, $r' = r' + x_m$ and $\Gamma' = \Gamma^*$ and repeat the procedure.

To complete the proof, it suffices to show that, the above steps are finite and in the final step, inequality (b) holds with equality or e_m is empty of flow. To see this, first observe that at any step (that ended without (b) being tight and without e_m being empty of flow), because of (I), the number of edges that have either $f'_e < o_e$ or ($f'_e = o_e$ and $\gamma'_e < \gamma$) drops down by at least one. In the worst case, this number will drop down to zero and thus in the next step either (b) will get tight or $x_m = o_{e_m}$, as then the rerouting may continue unrestricted. □

Next we give an upper bound of the PoA_γ of γ-modifiable instances with latency functions in class $\mathcal{D}$. The pair (f,Γ) of Lemma 7.2 is used in an analysis similar to [29]. The notable difference comes to the factor $-\gamma(x-y)d(x)$ that arises in $\sup_{d\in\mathcal{D},x\geq y\geq 0} \frac{y(d(x)-d(y))-\gamma(x-y)d(x)}{xd(x)}$ because of the ability to γ-modify the instances.

Theorem 7.3. For γ-modifiable parallel-links instances with cost functions in class $\mathcal{D}$, it is

$$PoA_\gamma(\mathcal{D}) \leq \rho_\gamma(\mathcal{D}) = \max\left\{1, \frac{1}{1-\beta_\gamma(\mathcal{D})}\right\},$$

where $\beta_\gamma(\mathcal{D}) = \sup_{d\in\mathcal{D},x\geq y\geq 0} \frac{y(d(x)-d(y))-\gamma(x-y)d(x)}{xd(x)}$

Proof. Let $\mathcal{G} = (G, r)$ be a parallel links instance with cost functions in class $\mathcal{D}$ and o be its optimal solution. For $\mathcal{G}$, Lemma 7.2 holds. Let f and Γ be the flow and the γ-modification given by Lemma 7.2. By definition, it is $PoA_\gamma(\mathcal{G}) \leq PoA(\mathcal{G}^\Gamma)$. We will bound $PoA(\mathcal{G}^\Gamma)$.

Let E^f_{max} be the set of edges that have $f_e < o_e$, E^f_{min} be the set of edges that have $f_e > o_e$ and E^f_{int} be the set of used edges with $f_e = o_e$. Clearly $E = E^f_{max} \cup E^f_{min} \cup E^f_{int}$

It is $PoA(\mathcal{G}^\Gamma) = \frac{\sum_{e\in E} f_e d_e(f_e)}{\sum_{e\in E} o_e d_e(o_e)}$. Using variational inequality and the γ_e's of Γ (for which we have a Nash flow) we get

$$\sum_{e\in E^f_{max}} f_e d_e(f_e) + \sum_{e\in E^f_{min}} f_e(1+\gamma) d_e(f_e) + \sum_{e\in E^f_{int}} f_e(1+\gamma_e) d_e(f_e)$$
$$\leq \sum_{e\in E^f_{max}} o_e d_e(f_e) + \sum_{e\in E^f_{min}} o_e(1+\gamma) d_e(f_e) + \sum_{e\in E^f_{int}} o_e(1+\gamma_e) d_e(f_e)$$

[4] induction hypothesis for the instance (G_m, r) could also be applied

$$\Updownarrow$$

$$\sum_{e\in E} f_e d_e(f_e) \leq \sum_{e\in E} o_e d_e(f_e) - \sum_{e\in E^f_{min}} \gamma(f_e - o_e) d_e(f_e) - \sum_{e\in E^f_{int}} \gamma_e(f_e - o_e) d_e(f_e)$$

$$\leq \sum_{e\in E} o_e(d_e(f_e) - d_e(o_e)) + \sum_{e\in E} o_e d_e(o_e) - \sum_{e\in E^f_{min}} \gamma(f_e - o_e) d_e(f_e)$$

$$\leq \sum_{e\in E} o_e d_e(o_e) + \sum_{e\in E^f_{min}} \Big(o_e(d_e(f_e) - d_e(o_e)) - \gamma(f_e - o_e) d_e(f_e)\Big)$$

Setting $\beta_\gamma(\mathcal{D}) = \sup_{d\in\mathcal{D}, x\geq y\geq 0} \frac{y(d(x)-d(y))-\gamma(x-y)d(x)}{xd(x)}$ we get

$$\sum_{e\in E} f_e d_e(f_e) \leq \sum_{e\in E} o_e d_e(o_e) + \sum_{e\in E^f_{min}} \frac{\Big(o_e(d_e(f_e) - d_e(o_e)) - \gamma(f_e - o_e) d_e(f_e)\Big) f_e d_e(f_e)}{f_e d_e(f_e)}$$

$$\leq \sum_{e\in E} o_e d_e(o_e) + \beta_\gamma(\mathcal{D}) \sum_{e\in E^f_{min}} f_e d_e(f_e)$$

$$\Downarrow$$

$$\frac{\sum_{e\in E} f_e d_e(f_e)}{\sum_{e\in E} o_e d_e(o_e)} \leq \frac{1}{1-\beta_\gamma(\mathcal{D})} = \rho_\gamma(\mathcal{D})$$

Putting everything together, we have

$$PoA_\gamma(\mathcal{G}) \leq PoA(\mathcal{G}^\Gamma) = \frac{\sum_{e\in E} f_e d_e(f_e)}{\sum_{e\in E} o_e d_e(o_e)} \leq \rho_\gamma(\mathcal{D})$$

Thus, for the class of games with latency functions in class $\mathcal{D}$ it is

$$PoA_\gamma(\mathcal{D}) \leq \rho_\gamma(\mathcal{D})$$

□

Next we give upper bounds for specific classes of γ-modifiable instances.

Corollary 7.4. For γ-modifiable instances with degree d polynomial latency functions

$$PoA = 1, \text{ if } \gamma \geq d \text{ and } PoA \leq \frac{1}{1 - d\Big(\frac{\gamma+1}{d+1}\Big)^{\frac{d+1}{d}} + \gamma}, \text{ if } \gamma < d$$

Proof. Let $\mathcal{G}$ be a γ-modifiable instance with degree d polynomial latency functions and let o be the optimal solution of $\mathcal{G}$.

By (e.g.) [77] o is the Nash flow of a game with the same underlying network and traffic rate r but cost functions changed from $d_e(x)$ to $(xd_e(x))'$. Thus, for polynomials of degree d, o coincides with the Nash flow of a game with the same underlying network and rate r but cost functions changed from $\sum_{i=0}^{d} a_i x^i$ to $\sum_{i=0}^{d}(i+1)a_i x^i$.

If $\gamma \geq d$, it is $\sum_{i=0}^{d}(i+1)a_i x^i \leq (\gamma+1)\sum_{i=0}^{d} a_i x^i$ and thus there exist γ_e's so as for any edge e $\sum_{i=0}^{d}(i+1)a_i o_e^i \leq (1+\gamma_e)\sum_{i=0}^{d} a_i o_e^i$, which implies that o is d-enforceable.

For $\gamma < d$ we bound $\beta_\gamma(\mathcal{D})$. It is $\frac{y(d(x)-d(y))-\gamma(x-y)d(x)}{xd(x)} = \frac{y}{x}\Big(1+\gamma-\frac{d(y)}{d(x)}\Big)-\gamma$. For a polynomial of degree d, $d(x) = \sum_{i=0}^{d} a_i x^i$, and $y \leq x$ it is $\frac{d(y)}{d(x)} \geq \frac{y^d}{x^d}$ as $y^{d-i} \leq x^{d-i} \Rightarrow \sum a_i y^i x^d \geq \sum a_i x^i y^d$. Thus $\frac{y}{x}\Big(1+\gamma-\frac{d(y)}{d(x)}\Big)-\gamma \leq \frac{y}{x}\Big(1+\gamma-\frac{y^d}{x^d}\Big)-\gamma$. This quantity is maximized for y satisfying $(y^{d+1})' = (1+\gamma)x^d$ which gives $y = \sqrt[d]{\frac{\gamma+1}{d+1}}x$, $\beta_\gamma(\mathcal{D}) = d\Big(\frac{\gamma+1}{d+1}\Big)^{\frac{d+1}{d}} - \gamma$ and $\rho_\gamma(\mathcal{D}) = \frac{1}{1-d\Big(\frac{\gamma+1}{d+1}\Big)^{\frac{d+1}{d}}+\gamma}$ □

By setting $d = 1$ to Corollary 7.4 we bound the price of anarchy for affine latencies.

Corollary 7.5. For γ-modifiable instances with affine latencies

$$PoA = 1, \text{ if } \gamma \geq 1 \text{ and } PoA \leq \frac{1}{1-\frac{(1-\gamma)^2}{4}}, \text{ if } \gamma < 1$$

Remark 7.6. Relating $\rho(\mathcal{D})$ (from [29]) with $\rho_\gamma(\mathcal{D})$ for the class $\mathcal{D}$ of affine latencies we get that $\rho_\gamma(\mathcal{D}) = \frac{1}{1+\frac{\gamma(2-\gamma)}{3}}\rho(\mathcal{D})$ which drops down quickly as γ grows from 0 to 1.

Next we show that these bounds are essentially tight.

Theorem 7.7. For any class of latency functions $\mathcal{D}$, and any $\epsilon > 0$ there is an instance $\mathcal{G}$ with latency functions in $\mathcal{D}$ with $PoA_\gamma(\mathcal{G}) \geq \rho_\gamma(\mathcal{D}) - \epsilon$

Proof. Let $\epsilon > 0$ and consider a γ-modifiable instance $\mathcal{G}$ of two parallel links, e_1 and e_2, flow rate r and cost functions: an arbitrary cost function $d_1(x) = d_\epsilon(x)$ (to be fixed later) in class $\mathcal{D}$ for link e_1 and the constant function $l_2(x) = (1+\gamma)d_\epsilon(r)$ for link e_2.

Under any γ-modification $\Gamma = (\gamma_1, \gamma_2)$, at Nash flow f of $\mathcal{G}^\Gamma$, all the flow goes through link e_1 and thus $SC^f = rl_\epsilon(r)$. At optimal flow o, let o_1 be the flow that goes through link e_1. It is $SC^o = (r-o_1)(\gamma+1)d_\epsilon(r) + o_1 d_\epsilon(o_1)$ and

$$PoA = \frac{rd_\epsilon(r)}{(r-o_1)(\gamma+1)d_\epsilon(r) + o_1 d_\epsilon(o_1)} = \frac{1}{1-\frac{o_1(d_\epsilon(r)-d_\epsilon(o_1))-\gamma(r-o_1)d_\epsilon(r)}{rd_\epsilon(r)}}$$

Flow o is optimal and thus $o_1 \in [0, r]$ is exactly the value that minimizes social cost and eventually maximizes $\frac{rd_\epsilon(r)}{(r-o_1)(\gamma+1)d(_\epsilon r)+o_1 d_\epsilon(o_1)}$ and so maximizes

$$\frac{o_1(d_\epsilon(r) - d_\epsilon(o_1)) - \gamma(r - o_1)d_\epsilon(r)}{rd_\epsilon(r)} \tag{7.1}$$

Recall that $\beta_\gamma(\mathcal{D}) = \sup_{d\in\mathcal{D},x\geq y\geq 0} \frac{y(d(x)-d(y))-\gamma(x-y)d(x)}{xd(x)}$ and let $d_\epsilon(x)$ be a cost function such that $\sup_{x\geq y\geq 0} \frac{y(d_\epsilon(x)-d_\epsilon(y))-\gamma(x-y)d_\epsilon(x)}{xd_\epsilon(x)} \geq \beta_\gamma(\mathcal{D}) - \frac{\epsilon(1-\beta_\gamma(\mathcal{D}))^2}{1-\epsilon(1-\beta_\gamma(\mathcal{D}))}$. Because o_1 maximizes (7.1), it is $\frac{o_1(d_\epsilon(r)-d_\epsilon(o_1))-\gamma(r-o_1)d_\epsilon(r)}{rd_\epsilon(r)} \geq \beta_\gamma(\mathcal{D}) - \frac{\epsilon(1-\beta_\gamma(\mathcal{D}))^2}{1-\epsilon(1-\beta_\gamma(\mathcal{D}))}$ and thus

$$PoA \geq \frac{1}{1 - \beta_\gamma(\mathcal{D}) + \frac{\epsilon(1-\beta_\gamma(\mathcal{D}))^2}{1-\epsilon(1-\beta_\gamma(\mathcal{D}))}} = \rho_\gamma(\mathcal{D}) - \epsilon$$

□

Based on Lemma 7.2, we present an algorithmic approach that given a γ-modifiable instance $\mathcal{G}$ computes a γ-modification Γ such that the Nash flow f of $\mathcal{G}^\Gamma$ is such that $\frac{SC^f}{SC^o} \leq \rho(\mathcal{D})$, with $\mathcal{D}$ being the class of latency functions of $\mathcal{G}$.

Lemma 7.8. Let o be the optimal flow of a parallel links instance $\mathcal{G}$. A flow f and a γ-modification Γ of $\mathcal{G}$ with the properties of Lemma 7.2 can be computed in time $\frac{|E|(|E|+1)}{2} Time(AlgPar)$, where $Time(AlgPar)$ denotes the time complexity of an algorithm $AlgPar$ that computes Nash flows in parallel links networks.

Proof. We first compute the optimal solution o of $\mathcal{G}$ and delete all unused by o edges, so we assume that w.l.o.g. all edges of G are used by o. Then, we search efficiently in the space of flows for an f that can be combined with a γ-modification Γ of $\mathcal{G}$ such that f and Γ satisfy Lemma 7.2. The key point we use from Lemma 7.2 (and prove later on) is that for the pair (f, Γ) of the proposition, the edges that have $f_e < o_e$ are the most costly used edges under o while the edges that have $f_e > o_e$ are the less costly edges under o.

In detail, for a combination $(k, l) : k + l \leq |E|$ find sets $E_{max} : |E_{max}| = k$ and $E_{min} : |E_{min}| = l$ such that E_{max} contains the k most costly edges of G under o and E_{min} contains the l less costly edges of G under o[5]. Let $\mathcal{G}'$ be the γ-modifiable instance that has only the edges E_{max} and E_{min}, traffic rate $r' = \sum_{e\in E_{max}\cup E_{min}}(o_e)$ and the latency functions of edges in E_{min} changed from $d_e(x)$ to $(1 + \gamma)d_e(x)$.

[5]so as $E_{max} \cap E_{min} = \emptyset$

For all combinations of $(k, l) : k + l \leq |E|$, we compute the Nash flow f' of $\mathcal{G}'$. If f' is such that (1) $\forall e \in E_{max} : f'_e \leq o_e$, (2) $\forall e \in E_{min} : f'_e \geq o_e$ and (3) $\forall e \in E \setminus (E_{max} \cup E_{min}) : \frac{1}{1+\gamma} L(\mathcal{G}') \leq d_e(o_e) \leq L(\mathcal{G}')$ then we let for each edge $e \in E \setminus (E_{max} \cup E_{min})$, $\gamma_e = \frac{L(\mathcal{G}')}{d_e(o_e)} - 1$. Clearly for any such e it is $0 \leq \gamma_e \leq \gamma$. By setting $\gamma_e = \gamma$ for all edges in E_{min} and $\gamma_e = 0$ for all edges in E_{min} we get a γ-modification Γ such that the Nash flow f of $\mathcal{G}^\Gamma$ satisfies Lemma 7.2 and thus Theorem 7.3 applies.

It remains to show that there is a pair (k, l) such that (1), (2) and (3) hold. By Lemma 7.2, there is a feasible flow f and a γ-modification Γ of $\mathcal{G}$ such that

i) f is the Nash flow in G^Γ.

ii) for any edge e: if $f_e < o_e$ then $\gamma_e = 0$ and if $f_e > o_e$ then $\gamma_e = \gamma$.

Let $E_{max} = \{e \in E | f_e < o_e\}$, $E_{min} = \{e \in E | f_e > o_e\}$ and $E_{int} = \{e \in E | f_e = o_e\}$. Because f is a Nash flow in G^Γ, for any $e_{max} \in E_{max}, e_{min} \in E_{min}, e_{int} \in E_{int}$ it is $d_{e_{max}}(f_{e_{max}}) \geq (1 + \gamma_{e_{int}}) d_{e_{int}}(f_{e_{int}}) \geq (1 + \gamma) d_{e_{min}}(f_{e_{min}})$. This, combined to [for $e \in E_{max}$ it is $d_e(o_e) \geq d_e(f_e)$, for $e \in E_{int}$ it is $d_e(o_e) = d_e(f_e)$ and for $e \in E_{min}$ it is $d_e(o_e) \leq d_e(f_e)$] implies that E_{max} contains the most costly edges under o and E_{min} contains the less costly edges under o. Thus for pair $(|E_{max}|, |E_{min}|)$, properties (1), (2) and (3) hold.

Clearly the above procedure needs at most $\frac{|E|(|E|+1)}{2}$ (possible pairs of (k, l)) computations of the Nash flow of G' under the different flow rates that may arise. Thus the lemma follows. □

7.3 Connection to Routing Games with Restricted Tolls

Bonifaci et al. in [17] studied routing games where along with each edge, an upper bound on the allowable toll on that edge is given. They provide a characterization for the flows that can be imposed by the restricted tolls and compute the optimal tolls when the optimal flow is inducible. We have drawn an iff condition (Theorem 7.1 for parallel links and a similar one for series parallel networks) that better suites our multiplicative approach and is also needed to follow the other results.

Also in [17], based on the previous characterization, they manage to compute the tolls that induce the smallest cost at equilibrium for parallel link networks. Their approach is essentially using a convex programming solver that solves several convex programs. One of these convex programs considers exactly the edges that is used by the best modification. This way though, no information about the nature of the solution is given.

Our approach intuitively applies Karush Kuhn Tucker conditions to the convex program that considers exactly the edges that is used by the

best modification and tries to follow them. It does so, by restricting the (f_e, γ_e) pairs on the edges in a way similar to the restrictions provided by the Karush Kuhn Tucker conditions (Lemma 7.2).

This approach has two major benefits. The first of them is that it relates the complexity of finding a good (wrt PoA bounds) γ-modification to the complexity of finding Nash flows. The second one is that it allows to get a simpler mathematical expression as a bound for the PoA. This expression is both giving a better insight for the improvement that can be achieved via γ-modifications[6] and applies to cases with more general latency functions (for which a tight example is given) than the polynomial latency functions considered in [17][7].

As a last positive comment for our work[8], we point out out that by our approach, uncertainty is applied only to edges that intuitively have to get some uncertainty while by the approach of [17] more or less all edges get some uncertainty which is the minimum between their toll bound or their marginal toll. Their derived bounds rely to the fact that in worst case examples there are edges of constant cost that take no toll. Thus in our approach the total amount of uncertainty is kept lower while for cases "away" from the worst case examples our modifications ought to give better improvement results.

7.4 Modifying Routing Games in more General Settings

In this section we show a way for generalizing the results of the parallel links case to the case of series parallel networks and to the case of parallel links with heterogeneous players and more general restrictions on the uncertainty added to each edge.

Series Parallel Networks

The main goal is to get a γ-modification similar to the one of Lemma 7.2 which restricts the pairs (f_e, γ_e) to be such that if $f_e > o_e$ then $\gamma_e = \gamma$ and if $f_e < o_e$ then $\gamma_e = 0$. Once such a Lemma is proved, the analysis of the PoA bound of Theorem 7.3 can be applied and a similar Theorem would hold.

[6] compare the PoA bound of γ-modifiable games with our approach, i.e. $\frac{1}{1-\beta_\gamma(\mathcal{D})}$, where $\beta_\gamma(\mathcal{D}) = \sup_{d\in\mathcal{D}, x\geq y\geq 0} \frac{y(d(x)-d(y))-\gamma(x-y)d(x)}{xd(x)}$, to the PoA bound of standard CGs (without tolls), i.e $\frac{1}{1-\beta(\mathcal{D})}$, where $\beta(\mathcal{D}) = \sup_{d\in\mathcal{D}, x\geq y\geq 0} \frac{y(d(x)-d(y))}{xd(x)}$.

[7] for general latency functions the approach of [17] is not guaranteed to give better bounds than the ones without tolls.

[8] for series parallel networks.

To prove such a Lemma, we use induction on the decomposition of the series parallel network. The complicated part of the induction is the parallel composing one. Assuming that for each of the parallel composing networks the induction hypothesis of the lemma holds we start rerouting flow from one network to the other in a careful way so as the properties for the (f_e, γ_e) pairs do not break. The way the rerouting occurs is similar to the rerouting in the parallel links case, i.e. all the γ_e's are locked and get unlocked only if f_e gets equal to o_e during the rerouting which itself reduces the largest fraction of used paths with common ends until it reaches $\gamma + 1$.

The constructive nature of this lemma will allow us also to get such a flow in time polynomially related to the time for computing a Nash flow, i.e. an analogue of Lemma 7.8 can be derived. The rerouting procedure will have to stop either to lock or unlock a γ_e of an edge e or to leave an edge empty of flow. Because each of these incidents may happen only once for each edge, we get that the rerouting will stop at most $O(|E|)$ number of times. For the exact flow that gets rerouted each time, a binary search in the space of flows has to be made, that calls several times a Nash flow solver, which at the end returns a flow for which one of the above incidents happens.

Parallel Links with Heterogeneous Players

The main goal is again to get a γ-modification similar to the one of Lemma 7.2. When players are heterogeneous then at equilibrium the more risk averse players get to higher expected cost edges. We assume that the edges are ordered according to their cost under optimal flow.

The new thing with the rerouting when trying to follow the proof of Lemma 7.2 is that it should stop also whenever a type of risk averse players entirely leaves an edge to get to the next one. If during the rerouting this happens, then the rerouting should continue up to that edge until this specific edge gets a cost equal to the cost of the next edge, where the cost is wrt the cost of the less risk averse users on this edge. That will allow the rerouting to continue to the next edges and the new player type to enter the next edge.

To compute such a flow-modification pair, a more sophisticated approach than the homogeneous case is needed. We know that in such a flow-modification pair only some of the highest cost (under optimal flow) edges, E_h, will get no uncertainty added, only some some of the lowest cost (under optimal flow) edges, E_l will get all the uncertainty they could and all other edges, E_i, will have $f_e = o_e$. For each combination of possible (E_h, E_i, E_l) triples, we do a binary search to see the amount of flow that will

go through E_h (which will directly give the amount of flow that goes through E_l) in order to have a (Nash flow,γ-modification) pair with the desired properties. For some triples we will have to stop the procedure as there may be no such pair for them, though, because of the previous lemma it ought to be a triple for which such a pair exists and which we eventually will find.

To derive a bound on the PoA we may follow a variational inequality approach, one for each risk aversion type. Then all these variational inequalities can be combined but with the risk averse factor on each edge being the lowest one on that edge. Thus in total a PoA bound (that is in fact tight) similar to Theorem 7.3 can be derived for the smallest risk averse factor among the players.

The above approach was allowing γ-modification on the edges. Another way to see γ-modifications is to allow γ_e's on the edges so as $\|(\gamma_e)_{e\in E}\|_\infty \leq \gamma$. So to generalize the approach we may let γ_e's to be such that $\|(\gamma_e)_{e\in E}\|_p \leq \gamma$ for any chosen p-norm. We then can show that treating such a p-norm allowable modification as in a $(\gamma/\sqrt[p]{|E|})$-modifiable CG, we can derive asymptotically tight results on the improvement of the PoA.

Chapter 8

Discussion

In the previous chapters we revealed our tiny contribution in the research arena, concerning Braess Paradox in bottleneck routing games, resolving Braess's paradox in random networks and studying CGs with uncertain delays and risk averse players. Yet, many things are open for research.

8.1 Braess' Paradox in Additive Costs Games

Research on Braess's Paradox for additive costs congestion games doesn't have many open fields, as there is a big portion of literature concerning it. One open problem that seems interesting to us is the one arose when resolving the paradox for random graphs.

Recall the steps we followed to establish the approximation algorithm. We first simplified the random instance we were given (the good whp network), then we approximately solved the simplified instance via an algorithm based on an approximate version of Caratheodory's Theorem ([12]) and finally we extended the result to the initial instance. Although the approximation factor relies both on the solution for the simplified instance and on the extension to the initial instance, the most costly part of the running time of the algorithm is the intermediate one where we solved the simplified instance.

To recall the problem see figure 8.1. Given a simplified instance, one should suitably remove some of the internal edges (edges from neighbors of s, N_s, to neighbors of t, N_t) so as to minimize, under traffic rate r, the Nash flow cost in the remaining network. Edges adjacent to s or t have affine latencies, i.e. of the form $ax + b$, while the internal edges have zero latency.

The problem can be formulated as a linear program with linearly com-

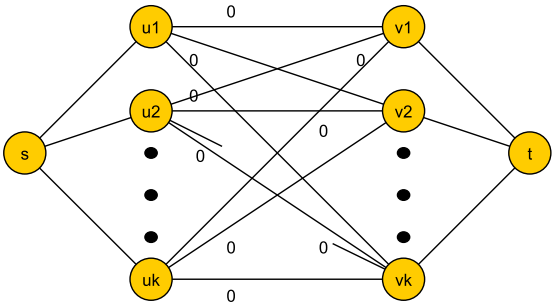

Figure 8.1: A simplified network. All edges adjacent to s or t have affine latency functions while all other edges, the internal edges, have zero latency

plementarity constraints (lplcc).

$$
\begin{aligned}
& & \textit{minimize } & \ l \\
s.t. & & a_i f_i + b_i &= l_i \quad \forall i \in N_s \\
& & a_j f_j + b_j &= l_j \quad \forall j \in N_t \\
& & \textstyle\sum_{i \in N_s} f_i &= r \\
& & \textstyle\sum_{j \in N_t} f_j &= r \\
& & \textstyle\sum_{j \in N_t} f_{ij} &= f_i \quad \forall i \in N_s \\
& & \textstyle\sum_{i \in N_s} f_{ij} &= f_j \quad \forall j \in N_t \\
& & f_{ij}(l - l_i - l_j) &= 0 \\
& & l_i, l_j, f_i, f_j, f_{ij} &\geq 0 \quad \forall i \in N_s, \forall j \in N_t
\end{aligned}
$$

Variables f_i, f_j, f_{ij} correspond to the flow that is routed through the i–th neighbor of s, u_i, the j–th neighbor of t, v_j, and edge $\{u_i, v_j\}$ respectively. Variables l_i and l_j correspond to the costs of edges $\{s, u_i\}$ and $\{v_j, t\}$ respectively, under flow f_i and f_j respectively. Variable f_{ij} corresponds to the flow through edge $\{u_i, v_j\}$.

The first two constraints capture the relation between the flow routed through edges neighboring to s or t and their corresponding costs. The next four constraints model flow conservation. The complementarity constraint ensures that if a path $s - u_i - v_j - t$ takes a positive flow, i.e. $f_{ij} > 0$, then its cost is l. If $f_{ij} = 0$ then we do not care about $l_i + l_j = l$ as we can remove edge $\{u_i, v_j\}$ in the subnetwork.

This formulation seems to tightly capture the problem. Solving general lplcc's is NP-hard but this case is not exactly of the form proved to be NP-hard as $l - l_i - l_j$ need not be non negative and also a feasible solution can be easily found (the equilibrium of the initial network satisfies the restrictions). Additionally, the size of the intermediate network, or more precisely, the many, long and correlated paths between s and t, is a key factor in the

NP-hardness reduction of [78]. Thus the complexity of finding the best subnetwork in simplified networks, even for linear latencies, remains well open (as far as we know) and seems quite intriguing.

8.2 Braess' Paradox in Bottleneck Costs Games

As we earlier saw, the inapproximability results of [52] did not really shed light on the approximability of the (simple, non-selective) network design problem in the simplest, and most interesting, setting of non-atomic bottleneck routing games with a common origin-destination pair for all players. In our work, concerning the paradox in bottleneck costs games, we followed this direction and managed to prove strong NP hardness results for detecting the paradox and approximating the best subnetwork.

In the model we studied, the players were considering the bottleneck cost of their paths and the social cost function was equal to the bottleneck cost of the network. Yet, there is another interesting variant of bottleneck games where the social cost is the average (instead of the maximum) bottleneck cost of the players, studied by Cole, Dodis, and Roughgarden [27] and subsequently by Mazalov et al. [62]. For this variant of non-atomic bottleneck congestion games, as far as we know, neither the problem of detecting the paradox nor the one of finding the best subnetwork has been studied. It feels like the techniques similar to the ones presented here could be used to prove results on these problems that seem to lay in between the problems for additive costs congestion games and the problems for bottleneck costs games with social cost function equal to the maximum congestion.

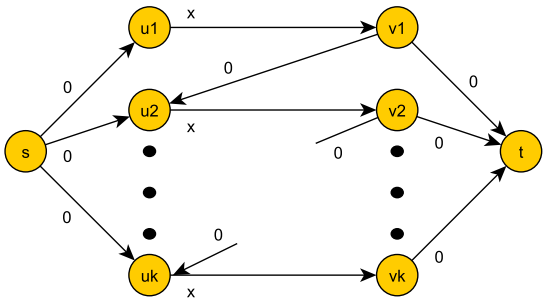

Figure 8.2: A network with $\Theta(n)$ PoA. Optimally all the flow is splitted equally to the k direct paths while a bad Nash flow routes the traffic through path $s - u_1 - v_1 - u_2 - v_2 - \ldots u_k - v_k - t$

Another research goal could be to derive results on the existence of

the paradox in random graphs under bottleneck costs. The analog of the additive costs case, i.e. to study random $\mathcal{G}_{n,p}$'s could be a first goal. The intuition suggests that a proof of existence of the paradox in random bottleneck costs networks should not be hard and that is because an analog of the bad example's "backedges" (edges going in "opposite" direction, from neighbors of t to neighbors of s) e.g. in [27] or figure 8.2 can probably be easily found in many models of random networks. In a bad Nash flow these backedges are used by the players in sequence and thus all paths from s to t cost much and the network behaves bad.

In addition to these, one could try using the techniques we used for exploiting the paradox in random graphs under additive costs so as to exploit the paradox. If specific expansion properties for the random graph models hold then this goal seems tractable and would be of interest.

8.3 Stochastic Congestion Games

While the above problems seem quite interesting, our focus is turned to stochastic congestion games and risk averse players. The natural models we proposed for the study of risk aversion under stochastic delays do not seem to easily "cooperate" with networks other than parallel links. This is not a problem that arises only in our models. The players' cost functions used in both models, do not have the property of being the sum of the costs of the edges on the players' paths, i.e. the costs are non-separable, and examining games with non separable costs is not an easy task. An interesting task would be to formulate players' behaviors in a "network familiar" way so as to be able to draw meaningful results for these games in general networks.

Staying in our model though still has some intriguing open problems. One of them is whether games with heterogeneous stochastic players admit an equilibrium. Our intuition is not clear. Another problem is to examine the case of games with stochastic edges and derive results on the existence of equilibrium and potential functions when networks are layered, with the basic property that all $s-t$ paths have the same length. In this case the paths' costs might be separable and we would be able to draw meaningful results for networks different from parallel links.

8.4 Abusing Uncertainty

In the direction of abusing uncertainty, we saw that when risk averse players act under uncertainty, a large portion of them might prefer choosing heavy in expectation but low variance links rather than light in expectation but high variance links. Adding some uncertainty to the network could help in improving its behavior. The results presented here hold for special cases of networks and homogeneous risk averse users.

There are two main open problems in this field. One is how can heterogeneousness in the players risk aversion help in improving the network's performance. In this direction we have (almost) proved that heterogeneousness can also help in improving the network's performance, yet in the worst case the improvement is similar to the one achieved when players are homogeneous.

The second and probably more interesting problem is how can we abuse uncertainty in more general networks under natral player's risk averse cost functions and draw meaningful results on improving the price of anarchy. When trying to analyze such cases the problem of non separable costs arises once more requiring risk averse cost functions that are both network familiar and capturing the averse behavior of the players.

Bibliography

[1] Heiner Ackermann, Heiko Röglin, and Berthold Vöcking, *On the impact of combinatorial structure on congestion games.*, 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 2006.

[2] _____, *Pure nash equilibria in player-specific and weighted congestion games.*, WINE (Paul G. Spirakis, Marios Mavronicolas, and Spyros C. Kontogiannis, eds.), Lecture Notes in Computer Science, vol. 4286, Springer, 2006, pp. 50–61.

[3] Heiner Ackermann and Alexander Skopalik, *Complexity of pure nash equilibria in player-specific network congestion games.*, Internet Mathematics **5** (2008), no. 4, 323–342.

[4] Sebastian Aland, Dominic Dumrauf, Martin Gairing, Burkhard Monien, and Florian Schoppmann, *Exact price of anarchy for polynomial congestion games*, STACS, 2006, pp. 218–229.

[5] Ingo Althöfer, *On sparse approximations to randomized strategies and convex combinations*, Linear Algebra and Applications **99** (1994), 339–355.

[6] H. Angelidakis, D. Fotakis, and T. Lianeas, *Stochastic congestion games with risk-averse players*, Proc. of the 6th International Symposium on Algorithmic Game Theory (SAGT '13), LNCS, vol. 8146, 2013, pp. 86–97.

[7] Itai Ashlagi, Dov Monderer, and Moshe Tennenholtz, *Resource selection games with unknown number of players*, Proc. of the 5th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 06), 2006, pp. 819–825.

[8] Baruch Awerbuch, Yossi Azar, and Amir Epstein, *The price of routing unsplittable flow*, Proceedings of the Thirty-seventh Annual ACM

Symposium on Theory of Computing (New York, NY, USA), STOC '05, ACM, 2005, pp. 57–66.

[9] Baruch Awerbuch, Yossi Azar, Amir Epstein, Vahab S. Mirrokni, and Alexander Skopalik, *Fast convergence to nearly optimal solutions in potential games.*, ACM Conference on Electronic Commerce (Lance Fortnow, John Riedl, and Tuomas Sandholm, eds.), ACM, 2008, pp. 264–273.

[10] Yossi Azar and Amir Epstein, *The hardness of network design for unsplittable flow with selfish users*, Proc. of the 3rd Workshop on Approximation and Online Algorithms (WAOA '05), LNCS, vol. 3879, 2005, pp. 41–54.

[11] Ron Banner and Ariel Orda, *Bottleneck routing games in communication networks*, IEEE Journal on Selected Areas in Communications **25** (2007), no. 6, 1173–1179.

[12] Siddharth Barman, *Approximating carathéodory's theorem and nash equilibria*, CoRR **abs/1406.2296** (2014).

[13] Martin Beckmann, C.B. McGuire, and Christopher B. Winsten, *Studies in the economics of transportation*, Yale University Press, New Haven (1956).

[14] Kshipra Bhawalkar, Martin Gairing, and Tim Roughgarden, *Weighted congestion games: Price of anarchy, universal worst-case examples, and tightness.*, ESA (2) (Mark de Berg and Ulrich Meyer, eds.), Lecture Notes in Computer Science, vol. 6347, Springer, 2010, pp. 17–28.

[15] Béla Bollobás, *Random Graphs, 2nd Ed.*, Cambridge Studies in Advanced Mathematics, no. 73, Cambridge University Press, 2001.

[16] Vincenzo Bonifaci, Tobias Harks, and Guido Schäfer, *Stackelberg routing in arbitrary networks*, Math. Oper. Res. **35** (2010), no. 2, 330–346.

[17] Vincenzo Bonifaci, Mahyar Salek, and Guido Schäfer, *Efficiency of restricted tolls in non-atomic network routing games*, Proceedings of the 4th International Conference on Algorithmic Game Theory (Berlin, Heidelberg), SAGT'11, Springer-Verlag, 2011, pp. 302–313.

[18] Costas Busch and Malik Magdon-Ismail, *Atomic routing games on maximum congestion*, **410** (2009), 3337–3347.

[19] Ioannis Caragiannis, Clemente Galdi, and Christos Kaklamanis, *Network load games*, Proc. of the 16th Symposium on Algorithms and Computation (ISAAC '05), LNCS, vol. 3827, 2005, pp. 809–818.

[20] Ioannis Caragiannis, Christos Kaklamanis, and Panagiotis Kanellopoulos, *Taxes for linear atomic congestion games*, ESA (Yossi Azar and Thomas Erlebach, eds.), Lecture Notes in Computer Science, vol. 4168, Springer, 2006, pp. 184–195.

[21] Steve Chien and Alistair Sinclair, *Convergence to approximate nash equilibria in congestion games*, Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (Philadelphia, PA, USA), SODA '07, Society for Industrial and Applied Mathematics, 2007, pp. 169–178.

[22] George Christodoulou and Elias Koutsoupias, *On the price of anarchy and stability of correlated equilibria of linear congestion games*,,, Proceedings of the 13th Annual European Conference on Algorithms (Berlin, Heidelberg), ESA'05, Springer-Verlag, 2005, pp. 59–70.

[23] George Christodoulou and Elias Koutsoupias, *The price of anarchy of finite congestion games*, Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing (New York, NY, USA), STOC '05, ACM, 2005, pp. 67–73.

[24] Fan Chung and Stephen J. Young, *Braess's paradox in large sparse graphs*, WINE (Amin Saberi, ed.), Lecture Notes in Computer Science, vol. 6484, Springer, 2010, pp. 194–208.

[25] Fan Chung, Stephen J. Young, and Wenbo Zhao, *Braess's paradox in expanders*, Random Struct. Algorithms **41** (2012), no. 4, 451–468.

[26] Richard Cole, Yevgeniy Dodis, and Tim Roughgarden, *Pricing network edges for heterogeneous selfish users*, STOC, ACM, 2003, pp. 521–530.

[27] _____, *Bottleneck links, variable demand, and the tragedy of the commons*, 2006, pp. 668–677.

[28] José R. Correa, Andreas S. Schulz, and Nicolás E. Stier Moses, *Selfish routing in capacitated networks*, Math. Oper. Res. **29** (2004), no. 4, 961–976.

[29] J.R. Correa, A.S. Schulz, and Nicolás Stier Moses, *Selfish Routing in Capacitated Networks*, Mathematics of Operations Research **29** (2004), no. 4, 961–976.

[30] Juliane Dunkel and Andreas S. Schulz, *On the complexity of pure-strategy nash equilibria in congestion and local-effect games*, In Proc. of the 2nd Int. Workshop on Internet and Network Economics (WINE, Springer Verlag, 2006, pp. 62–73.

[31] Amir Epstein, Michal Feldman, and Yishay Mansour, *Efficient graph topologies in network routing games*, Games and Economic Behaviour **66** (2009), no. 1, 115–125.

[32] Eyal Even-dar, Alex Kesselman, and Yishay Mansour, *Convergence time to nash equilibria*, In ICALP, Springer-Verlag, 2003, pp. 502–513.

[33] Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar, *The complexity of pure nash equilibria*, ACM Press, 2004, pp. 604–612.

[34] Amos Fiat and Christos H. Papadimitriou, *When the players are not expectation maximizers*, Proc. of the 3th International Symposium on Algorithmic Game Theory (SAGT '10), LNCS, vol. 6386, 2010, pp. 1–14.

[35] L. Fleischer, K. Jain, and M. Mahdian, *Tolls for Heterogeneous Selfish Users in Multicommodity Networks and Generalized Congestion Games*, 2004, pp. 277–285.

[36] Lisa Fleischer, Kamal Jain, and Mohammad Mahdian, *Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games*, FOCS, IEEE Computer Society, 2004, pp. 277–285.

[37] Steven Fortune, John E. Hopcroft, and James Wyllie, *The directed subgraph homeomorphism problem*, Theor. Comput. Sci. **10** (1980), 111–121.

[38] Dimitris Fotakis, *Stackelberg strategies for atomic congestion games*, ESA (Lars Arge, Michael Hoffmann, and Emo Welzl, eds.), Lecture Notes in Computer Science, vol. 4698, Springer, 2007, pp. 299–310.

[39] ———, *Congestion games with linearly independent paths: Convergence time and price of anarchy*, Theory of Computing Systems **47** (2010), no. 1, 113–136.

[40] Dimitris Fotakis, Alexis C. Kaporis, Thanasis Lianeas, and Paul G. Spirakis, *Resolving braess's paradox in random networks*, WINE (Yiling Chen and Nicole Immorlica, eds.), Lecture Notes in Computer Science, vol. 8289, Springer, 2013, pp. 188–201.

[41] ———, *On the hardness of network design for bottleneck routing games*, Theor. Comput. Sci. **521** (2014), 107–122.

[42] Dimitris Fotakis, Alexis C. Kaporis, and Paul G. Spirakis, *Efficient methods for selfish network design*, Proc. of the 36th Colloquium on Automata, Languages and Programming (ICALP-C '09), LNCS, vol. 5556, 2009, pp. 459–471.

[43] Dimitris Fotakis, George Karakostas, and Stavros G. Kolliopoulos, *On the existence of optimal taxes for network congestion games with heterogeneous users*, SAGT (Spyros C. Kontogiannis, Elias Koutsoupias, and Paul G. Spirakis, eds.), Lecture Notes in Computer Science, vol. 6386, Springer, 2010, pp. 162–173.

[44] Dimitris Fotakis, Spyros Kontogiannis, and Paul G. Spirakis, *Selfish Unsplittable Flows*, Theoretical Computer Science **348** (2005), 226–239.

[45] Dimitris Fotakis and Paul G. Spirakis, *Cost-balancing tolls for atomic network congestion games*, WINE (Xiaotie Deng and Fan Chung Graham, eds.), Lecture Notes in Computer Science, vol. 4858, Springer, 2007, pp. 179–190.

[46] Martin Gairing and Max Klimm, *Congestion games with player-specific costs revisited.*, SAGT (Berthold Vocking, ed.), Lecture Notes in Computer Science, vol. 8146, Springer, 2013, pp. 98–109.

[47] Fan Chung Graham, Stephen J. Young, and Wenbo Zhao, *Braess's paradox in expanders*, Random Struct. Algorithms **41** (2012), no. 4, 451–468.

[48] Torben Hagerup and Christine Rüb, *A guided tour of chernoff bounds*, Inf. Process. Lett. **33** (1990), no. 6, 305–308.

[49] Michael A. Hall, *Properties of the equilibrium state in transportation networks*, Transportation Sci. **12(3)** (1978), 208–216.

[50] Tobias Harks and Max Klimm, *On the existence of pure nash equilibria in weighted congestion games*, Mathematics of Operations Research **37** (2012), no. 3, 419–436.

[51] Martin Hoefer, Lars Olbrich, and Alexander Skopalik, *Taxing subnetworks.*, WINE (Christos H. Papadimitriou and Shuzhong Zhang, eds.), Lecture Notes in Computer Science, vol. 5385, Springer, 2008, pp. 286–294.

[52] Haiyang Hou and Guochuan Zhang, *The hardness of selective network design for bottleneck routing games*, Proc. of the 4th Conference on Theory and Applications of Models of Computation (TAMC '07), LNCS, vol. 4484, 2007, pp. 58–66.

[53] Tomas Jelinek, Marcus Klaas, and Guido Schäfer, *Computing Optimal Tolls with Arc Restrictions and Heterogeneous Players*, 31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014) (Dagstuhl, Germany) (Ernst W. Mayr and Natacha Portier, eds.), Leibniz International Proceedings in Informatics (LIPIcs), vol. 25, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2014, pp. 433–444.

[54] George Karakostas, *Edge pricing of multicommodity networks for heterogeneous selfish users*, In Proceedings of the 45th Annual Symposium on Foundations of Computer Science (FOCS, Press, 2004, pp. 268–276.

[55] George Karakostas and Stavros G. Kolliopoulos, *The Efficiency of Optimal Taxes*, Proc. of the 1st Workshop on Combinatorial and Algorithmic Aspects of Networking (CAAN '04), LNCS, vol. 3405, 2005, pp. 3–12.

[56] Frank Kelly, *The mathematics of traffic in networks*, In *The Princeton Companion to Mathematics* (Editors: T. Gowers, J. Green and I. Leader). Princeton University Press, 2008.

[57] Elias Koutsoupias and Christos Papadimitriou, *Worst-case equilibria*, *STACS* ***1563*** *(1999), 404-413.*

[58] Henry Lin, Tim Roughgarden, Éva Tardos, and Asher Walkover, Braess's Paradox, Fibonacci Numbers, and Exponential Inapproximability, LNCS, vol. 3580, 2005, pp. 497-512.

[59] Henry Lin, Tim Roughgarden, and Éva Tardos, A stronger bound on braess's paradox, Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms (Philadelphia, PA, USA), SODA '04, Society for Industrial and Applied Mathematics, 2004, pp. 340-341.

[60] Richard J. Lipton and Neal E. Young, Simple strategies for large zero-sum games with applications to complexity theory, Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada, 1994, pp. 734-740.

[61] Mavronicolas, Marios, Milchtaich, Igal, Monien, Burkhard, Tiemann, and Karsten, Congestion games with player-specific constants., MFCS (Ludek Kucera and Antonin Kucera, eds.), Lecture Notes in Computer Science, vol. 4708, Springer, 2007, pp. 633–644.

[62] Vladimir Mazalov, Burkhard Monien, Florian Schoppmann, and Karsten Tiemann, Wardrop equilibria and price of stability for bottleneck games with splittable traffic, LNCS, vol. 4286, 2006, pp. 331–342.

*[63] Igal Milchtaich, Congestion Games with Player-Specific Payoff Functions, Games and Economic Behavior **13** (1996), 111–124.*

*[64] ———, Network topology and the efficiency of equilibrium, Games and Economic Behavior **57** (2006), 321–346.*

[65] ———, Network topology and equilibrium existence in weighted network congestion games, 2009.

*[66] Dov Monderer and Lloyd Shapley, Potential Games, Games and Economic Behavior **14** (1996), 124–143.*

[67] Evdokia Nikolova and Nicolás Stier Moses, Stochastic selfish routing, Proc. of the 4th International Symposium on Algorithmic Game Theory (SAGT '11), LNCS, vol. 6982, 2011, pp. 314–325.

*[68] Fernando Ordóñez and Nicolás Stier Moses, Wardrop equilibria with risk-averse users, Transportation Science **44** (2010), no. 1, 63–86.*

*[69] Panagiota N. Panagopoulou and Paul G. Spirakis, Algorithms for pure nash equilibria in weighted congestion games., ACM Journal of Experimental Algorithmics **11** (2006).*

[70] Arthur C. Pigou, The economics of welfare, Macmillan, 1924.

[71] Georgios Piliouras, Evdokia Nikolova, and Jeff S. Shamma, Risk sensitivity of price of anarchy under uncertainty, Proceedings of the Fourteenth ACM Conference on Electronic Commerce (New York, NY, USA), EC '13, ACM, 2013, pp. 715–732.

[72] R. Tyrell Rockafellar, Coherent Approaches to Risk in Optimization Under Uncertainty, Tutorials in Operations Research (2007), 38–61.

[73] Robert W. Rosenthal, A class of games possessing pure-strategy nash equilibria, International Journal of Game Theory (1973), no. 2, 65–67.

[74] Tim Roughgarden, The price of anarchy is independent of the network topology, J. Comput. Syst. Sci. **67** (2003), no. 2, 341–364.

[75] ———, Stackelberg scheduling strategies, SIAM J. Comput. **33** (2004), no. 2, 332–350.

[76] ———, Selfish routing and the Price of Anarchy, MIT press, 2005.

[77] ———, Selfish Routing and the Price of Anarchy, MIT press, 2005.

[78] ———, On the severity of braess's paradox: Designing networks for selfish users is hard, J. Comput. Syst. Sci. **72** (2006), no. 5, 922–953.

[79] ———, Intrinsic robustness of the price of anarchy, Proceedings of the 41st Annual ACM Symposium on Theory of Computing (New York, NY, USA), STOC '09, ACM, 2009, pp. 513–522.

[80] Alan Siegel, Median bounds and their application, 1999, pp. 776–785.

[81] Alexander Skopalik and Berthold Vöcking, Inapproximability of pure nash equilibria, IN PROCEEDINGS OF THE 40TH ANNUAL ACM SYMPOSIUM ON THEORY OF COMPUTING (STOC, 2008, pp. 355–364.

[82] Chaitanya Swamy, The effectiveness of stackelberg strategies and tolls for network congestion games, SODA (Nikhil Bansal, Kirk Pruhs, and Clifford Stein, eds.), SIAM, 2007, pp. 1133–1142.

[83] Amos Tversky and Daniel Kahneman, Prospect theory: An analysis of decision under risk, Econometrica **47** (1979), no. 2, 263–291.

[84] Gregory Valiant and Tim Roughgarden, Braess's paradox in large random graphs, Random Struct. Algorithms. (Preliminary version in Proc. of EC'06) **37** (2010), no. 4, 495–515.

[85] László A. Végh, Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives, Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing, STOC '12, 2012, pp. 27–40.

[86] Hai Yang and Hai-Jun Huang, The multi-class, multi-criteria traffic network equilibrium and systems optimum problem, Transportation Research **38** (2004), 1–15.

List of Figures

Thesis proudly powered by LaTeX (and eBamp)