INFO4990 Research Proposal for the project "Game Theoretic Analysis of Internet Network Problems"

Enoch Lau
SID 200415765

4 June 2007

Abstract

In a network, uncooperative agents aim to minimise their own personal cost (such as congestion-related latency between source and destination) without regard to wider, societal impacts of their behaviour. It is well known that such selfish behaviour can result in large overall congestion, as studied through such measures as the price of anarchy, which is the ratio of the cost of the worst-case Nash equilibrium to that of the optimal flow. In this project, we aim to study the effect of turning the effects of congestion on its head – by taking congestion to be a desirable attribute, or by reciprocating the functions that define edge latencies. Motivating examples include the routing of traffic in a multicast fashion, and the study the behaviour of consumers attracted to items popular with other consumers. We aim to study a number of such models experimentally, and then extrapolate the results to theorems that can be proved theoretically. This research proposal will outline the motivation and context of the problem, the proposed contributions, a literature review, research methods and a research plan.

Contents

Chapter 1

Introduction

In this chapter, we consider the background to the selfish routing problem, and consider two motivating examples. We briefly outline the direction that the project is headed towards.

1.1 Background

The economic notions of game theory, as introduced by von Neumann and Morgenstern, intersected with theoretical computer science, finds fertile ground in the analysis of the behaviour of users on the Internet. As it is a network of wildly diverse users, some in competition and others in collaboration, a game theoretical analysis of the Internet can shed light on the inefficiencies caused by a lack of centralised control, and suggest economic measures that can be deployed to enforce a social optimum [14]. Congestion games were first studied by Rosenthal [15], but it is only recently that applications to networks have been explored. Further afield, for comparison, other aspects of game theory that impact on concepts in computer science include automated auctions, and mechanism design, where the aim is reversed, to create a game with the given outcomes [12].

1.2 Motivating Examples

The non-optimality of selfish routing has been known for some time. We will explore the two common motivating examples, Pigou's example and Braess' Paradox; the first was discovered in 1920, and the latter in 1968.

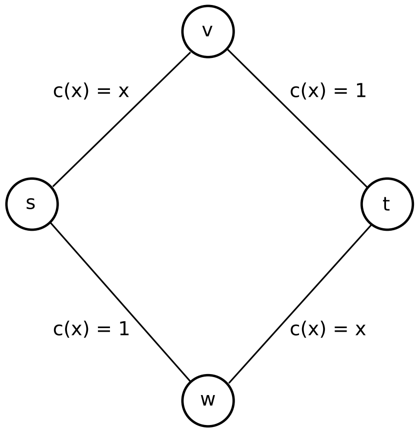

Figure 1.2: Braess' Paradox: the initial set-up

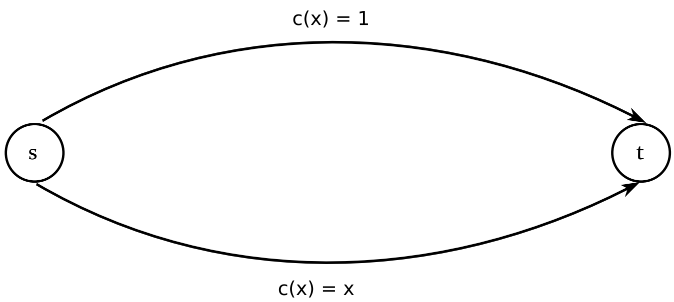

Figure 1.1: Pigou's example: a cost function $c(x)$ describes the latency experienced by users of the edge in question, as a function of the proportion of users on that edge.

In Pigou's example (Figure 1.1), there are two nodes, s and t, which we can conceptualise as towns, and two edges, or roads, that lead from s to t. Drivers want to drive from s to t in the shortest possible time. Each of the two roads has a cost function $c(\cdot)$ that represents the delay in say, hours, experienced by drivers using the road, as a function of the proportion of users using that particular road. It can be shown that the socially optimum behaviour (as measured by the average time taken by a driver) is when half the users take the upper road, and the other half take the lower road, leading to an average trip time of 45 minutes. However, selfish behaviour dictates that drivers on the upper road would want to switch to the lower road, which has a lower latency; this behaviour results in an average trip time of 1 hour. Pigou's simple example thus demonstrates that selfish behaviour need not produce a socially optimal outcome.

In Braess' Paradox, we obtain a counterintuitive result. We start with an initial network set-up as illustrated in Figure 1.2. Because the top and bottom paths are identical, the traffic is split evenly between the two paths, leading to an average trip time of 90 minutes. This is modified by adding an additional road with no latency to the network. This augmented set-up is illustrated in Figure 1.3.

All of the selfish users will take the route $svwt$, because going from s to t via any other route will take longer, resulting in an average trip time of 2 hours. Braess' Paradox thus illustrates a counter-

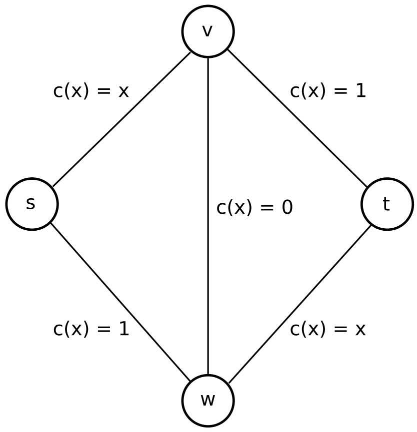

Figure 1.3: Braess' Paradox: the augmented network with the addition of the edge vw

intuitive situation, that network improvements have the ability to degrade network performance in the face of selfish users.

1.3 The Road Ahead

The surprising fact is that these simple examples often suffice to show the worst kinds of behaviour possible in these types of networks; for example, the price of anarchy in Pigou's example is the worst that is attainable for a certain class of networks [18]. Other work in the field, as will be explained in greater detail in Chapter 2, have analysed worst-case bounds for steeper cost functions, approximate equilibria and edge capacities to name a few.

However, this project will be primarily focused on examining cost functions that are the reciprocals of those functions already studied; in effect, the cost functions become decreasing in the number of users, and will thus reward congestion as opposed to penalising congestion. Distributing the cost in a direct inverse of the number of users has been studied as the Shapley value in the context of optimising multicast transmission [3], but no general study of reciprocal cost functions has been performed, and no study of inverse functions in a network setting other than multicast has been performed either. In another perspective to extending the selfish routing problem, this project generalises the work in [3], and then seeks to reapply the problem back to multicast routing to study the effects of new types of charging mechanisms.

The notion of a reciprocal congestion function has natural applications, in situations where congestion (a large number of users choosing the same strategy) is desired. For example, multicast transmission over the Internet differs from traditional unicast in that data being streamed to a large number of users is not repeated over the same link; in fact, one could see that if more users share the same transmission, the per user cost may be decreased. Other applications include the ability to buy in bulk if a large number of consumers purchase the same product, or the decrease in support cost in a less-diverse ecosystem of software products. There are of course, limits to congestion, and these can be modelled as edge capacities; limits to benefits of congestion in real life include the loss of the benefits of diversity, and the finite amount of a particular resource.

1.4 Definitions

1.4.1 The Model

We first describe a network model that has been used as the basis for several results about the price of anarchy, as formulated in [17]. In a network G, there are source-destination pairs $\{s_i, t_i\}$, called *commodities*, between which the network must route r_i units of traffic. For any commodity i, there may be several *paths* to choose from, and traffic may be split between these paths. *Flow* is the aggregation of the routes chosen by a large number of agents, and each agent is assumed to control a negligible amount of flow. A flow is said to be *feasible* if for all i, all required units of traffic are routed, that is:

$$\sum_{P \in \mathscr{P}_i} f_P = r_i \tag{1.1}$$

Edges have an associated *cost function* $c_e(\cdot)$ that may depend on the amount of flow on the edge, and the cost of a path P is denoted by c_P. If f_P is the amount of agents using path P, and f_e is the amount of flow using paths that include the edge e, then we can say that the total cost of a flow f is:

$$C(f) = \sum_{P \in \mathscr{P}} c_P(f) f_P = \sum_{e \in E} c_e(f_e) f_e \tag{1.2}$$

An *instance* is a triple of the form (G, r, c).

1.4.2 Nash Equilibrium

In general, a game consists of n players, and player i may choose from a set of strategies S_i [14]. Each player has a payoff function $u_i : S_1 \times \cdots \times S_n \to \mathbb{R}$ that measures the amount that player i obtains given the strategies chosen by the other players as well. Rational behaviour is deemed to be one in which no player has incentive to deviate, and this is the Nash equilibrium. Formally, it is a combination of strategies $x_1 \in S_1, \ldots, x_n \in S_n$ for which $u_i(x_1, \ldots, x_i, \ldots, x_n) \geq u_i(x_1, \ldots, x_i', \ldots, x_n)$ for all i and $x_i' \in S_i$. Specifically, flows at Nash equilibrium are flows such that if an agent were to switch paths, its cost would only increase.

1.4.3 Price of Stability and Price of Anarchy

The *price of stability* is defined as the ratio between the best flow at Nash equilibrium and that of an optimal flow [8]. This has a natural interpretation, as game theory, in particular, mechanism design, was used early on to design strategies in games to create certain desired outcomes.

With structures like the Internet, however, where the network was not designed, but instead evolved, a different measure of the anarchy that would result from the lack of centralised control was needed. The *price of anarchy* is defined as the worst-case ratio between the cost of a flow at Nash equilibrium and that of an optimal flow [18, 7]:

$$\rho(G, r, c) = \frac{C(f)}{C(f^*)} \tag{1.3}$$

Chapter 2

Literature Review

This is a review of the literature in this field of selfish routing, and the selfish routing model as applied to the multicast routing problem. We begin by looking at the bounds on the price of anarchy for various families of cost functions, and for various types of flows (splittable and unsplittable), that have been discovered in the literature. Secondly, we examine proposals for charging users for the user of a multicast stream that alleviate the inefficiencies posed by selfish routing, as one of our motivating applications of this field. Thirdly, we consider the types of games where a Nash equilibrium exists, and the computational complexity of computing them, as their study would fruitless if they did not exist, or could not be found easily. Finally, we summarise some of the notable techniques used in some of the proofs in the literature.

2.1 Results on the Price of Anarchy

2.1.1 Bounds on the Price of Anarchy

On the traffic model as defined in the definitions section in Chapter 1, [18] proves that for edge cost functions of the form $c_e(x) = a_e(x) + b_e$ for some $a_e, b_e \geq 0$, the price of anarchy is:

$$\rho(G, r, c) \leq \frac{4}{3} \tag{2.1}$$

This fundamental result cannot be improved, as demonstrated by Pigou's Example, which has a price of anarchy of $4/3$.

Further from linear edge cost functions, the paper then demonstrates that if $\mathscr{I}_p$ is the set of instances with cost functions that are polynomials up to a degree of p, then the price of anarchy is:

$$\sup_{(G,r,c)\in\mathscr{I}_p} \rho(G, r, c) = \Theta\left(\frac{p}{\ln p}\right) \tag{2.2}$$

However, for more general sets of functions, we introduce the concept of the anarchy value, which, for a set, captures how ill-behaved a set of allowable functions is. If c is a cost function, the *anarchy value* of c is defined, in [17], to be:

$$\alpha(c) = \sup_{x,r\geq 0} \frac{r \cdot c(r)}{x \cdot c(x) + (r - x)c(r)} \tag{2.3}$$

If $\mathscr{C}$ is a set of cost functions, then the *anarchy value* of $\mathscr{C}$ is:

$$\alpha(\mathscr{C}) = \sup_{0 \neq c \in \mathscr{C}} \alpha(c) \tag{2.4}$$

This is useful for characterising the price of anarchy for general sets of allowable functions. If $\mathscr{C}$ is a set of cost functions with anarchy value $\alpha(\mathscr{C})$, and (G, r, c) is an instance with cost functions in $\mathscr{C}$, then:

$$\rho(G, r, c) \leq \alpha(\mathscr{C}) \tag{2.5}$$

In general, this can be unbounded; that is, the cost of the Nash equilibrium can be arbitrarily worse than that of the optimal flow.

2.1.2 Bicriteria Bound in General Networks

[18] proves that if f is a flow at Nash equilibrium for (G, r, c), and f^* is feasible for $(G, 2r, c)$, then $C(f) \leq C(f^*)$; that is, a flow at Nash equilibrium is no worse than an optimal flow forced to route twice as much traffic.

2.1.3 Marginal Cost Function

[17] defines a *marginal cost function* (which captures the cost for new flow, as well as the additional congestion costs for existing traffic) for a differentiable function c:

$$c^* = \frac{d}{dx}(x \cdot c(x)) = c(x) + x \cdot c'(x) \tag{2.6}$$

The work shows that a flow f feasible for an instance (G, r, c) is optimal iff it is at Nash equilibrium for the instance (G, r, c^*), a result which relates the optimal and Nash flows.

2.1.4 Variations on the Traffic Model

Non-atomic Congestion Games

Non-atomic congestion games (NCGs) are a generalisation of the above traffic model onto games that do not involve networks. [19] demonstrates that many of the above results carry through to NCGs. This result is interesting, because the non-atomic congestion game is the limit of the general congestion game as n, the number of players, goes to infinity.

Approximate Nash Flows

A flow f feasible for an instance (G, r, c) is at ϵ-approximate Nash equilibrium if the cost of a path is smaller than a factor of $(1+\epsilon)$ of any other path for that commodity. [18] extends the previous result for networks with linear cost functions to show that the price of anarchy is (tightly) bounded from above by:

$$C(f) \leq \frac{4+4\epsilon}{3-\epsilon} \cdot C(f^*) \tag{2.7}$$

Atomic Selfish Routing

In comparison to the traffic routing model, we now have a finite number of agents, each of which controls a non-negligible amount of flow. There are two cases:

- Splittable flow:
  - The difference between this case and the traffic routing model is that in the atomic splittable case, the agents will not take into account the congestion it causes for its own traffic.
  - In [18], it was shown that the bicriteria bound and the upper bound on the price of anarchy both extend to the atomic splittable case.
- Unsplittable flow:
  - In [18], the bicriteria bound and the upper bound on the price of anarchy both do not extend to the atomic unsplittable case.
  - However, the *best* Nash flow obeys both, subject to certain constraints regarding the cost functions and all commodities share the same source vertex).
  - In [2], the authors were able to further this work and find expressions for the price of anarchy for the general network case for unsplittable flow.
    * For linear latency functions, for weighted demand, the price of anarchy is $\frac{3+\sqrt{5}}{2}$ for pure and mixed strategies, and for unweighted demand, the price of anarchy is 2.5 for pure strategies only.
    * For polynomials of degree d latency, the price of anarchy is $d^{\Theta(d)}$ for pure and mixed is bounded from above by $O(2^d d^{d+1})$ and from below by $\Omega(d^{d/2})$.

2.2 Charging for Multicast

2.2.1 Multicast

Traditional Internet routing is based on unicast, where for each receiver, one stream of data is sent from the source. This can result in duplication of data, especially when the same information is sent down the same link twice. Multicast alleviates this problem by building a tree, typically to minimise the multicast rate or to minimise the multicast cost. When confronted with self-interested agents, the selfish multicast problem is one of designing economic incentives so that self-interested participants are guided to behave in a way that leads to the social optimum. Crucially, the difference between selfish multicast routing and the selfish routing model presented above is that in this case, per-user costs for an edge are typically non-increasing in the number of users of the edge.

2.2.2 Selfish Users

[3] models a selfish multicast game in which the cost of each edge is split evenly between those who use it; this is based on the Shapley value [20]. Players are added one-by-one into the network, and they use best response dynamics to pick the path to the source. There are two cases top consider, the integral (unsplittable) multicast case, and the fractional (splittable) multicast case.

Integral Multicast Game

The paper presents a proof that the price of anarchy of a Nash equilibrium obtained from best response dynamics has an upper bound of:

$$O(\sqrt{n}\log^2 n) \tag{2.8}$$

and a lower bound of:

$$\Omega\left(\frac{\log n}{\log\log n}\right) \tag{2.9}$$

Fractional Multicast Game

The price of anarchy of a Nash equilibrium in this case has an upper bound of:

$$O(\log n) \tag{2.10}$$

2.2.3 Incorporating Receivers' Utilities

In [5], we have the added consideration of factoring the users' utilities, u, of participating in the multicast transmission. A cost-sharing mechanism is a pair of formulas that calculate $x(u)$, the cost that an agent or user has to pay for participation in the network, and $\sigma(u)$, a binary vector that indicates whether a user is included or not. There are several desirable properties for the choice of mechanism:

- Strategyproofness
- No Positive Transfers
- Voluntary Participation
- Consumer Sovereignty
- Budget-Balance
- Efficiency

However, there are complications caused by combinations of these properties; for example, there is no strategyproof mechanism that is both budget-balanced and efficient [10]. These considerations suggest only the Marginal Cost mechanism, which is based on the marginal contribution to the overall welfare provided by a user having non-zero utility for the transmission, and the Shapley Value mechanism [20], which is based on sharing the cost of a link by those users downstream from the link. The Marginal Cost mechanism needs only a linear number of messages, while the Shapley Value mechanism requires a quadratic number of messages for communication.

2.2.4 Selfish Information Flows

Selfish information flows, as described by [1], assumes that each part of the flow (which would correspond to a single packet) attempts to minimise its own cost as it travels through edges of varying weights, instead of minimising the cost from the users' perspective. In [8], two cases were studied for selfish information flows: the uncapacitied and capacitied networks.

Uncapacitied Networks

The Shapley value-based cost sharing does not enforce min-cost multicast flows, because the Shapley value is inherently a local cost sharing method. Instead, a pair of path-based LPs were formulated, assuming the unique replicable and encodable properties of information flows. The solution of the LP gives the shadow prices, which intuitively measure the amount that the objective function will grow if a constraint is relaxed by one unit. The shadow price method guarantees a price of stability of 1, but the price of anarchy is only 1 in specific examples, for example, where we adopt a two-tier charging scheme.

Capacitied Networks

The LP is modified to include constraints relating to finite edge capacities, and a vector t in the dual, that represents the taxes on edges. The addition of taxes, which can then be refunded, guides the formation of an equilibrium, and results in a strategyproof system, where edges have no incentive to lie about their cost. After the taxes are removed, the multicast flow is still at Nash equilibrium, and budget-balance is still maintained.

2.3 Existence and Computational Complexity of Nash Equilibria

2.3.1 Classical Results

Nash proved that all games have mixed Nash equilibria [11]. His proof was based on Kakutani's fixed point theorem [6], which is a generalisation of Brouwer's fixed point theorem. The proofs were all non-constructive proofs, and did not give any insight into how one might construct such equilibria [13].

Closer to the selfish routing problem, Rosenthal, in a seminal paper [15], proved that non-cooperative games, that is, congestion games, always have pure Nash equilibria. Monderer and Shapley [9] proved that any exact potential game (games are associated with a potential function that guides the players towards equilibrium) is isomorphic to a congestion game. However, [4] shows that this requirement that the differences be equal when considering potential functions is too strict, and argues that using *general potential functions*, where all that matters is the sign, is valid.

2.3.2 Polynomial-Time Algorithms in Restricted Classes

[4] shows that there was a polynomial time algorithm for calculating a pure Nash equilibrium in the symmetric network case, that is, the case where all players have the same source and destination. The paper also shows that it is possible to approximate the Nash equilibria of non-atomic congestion games in strongly polynomial time.

2.3.3 Complexity of Calculating a Nash Equilibrium in the General Case

The difficulty with finding Nash equilibria is that their proof is non-constructive. [4] proves that computing a Nash equilibrium in the general network case is PLS-complete (which, intuitively, corresponds to "as hard to compute as any object whose existence is guaranteed by a potential function").

2.3.4 Existence of Nash Equilibria for the Selfish Multicast Problem

In [3], it was noted that the integral mutlicast game is, by definition, a congestion game, and thus by Rosenthal, has a pure Nash equilibrium. For the fractional multicast game, the potential function is:

$$\Phi = \sum_{e \in s} \left(\sum_{j=1}^{n_e(s)} \sum_{i=1}^{n_e+1-j} c_e \frac{f_{e,j} - f_{e,j-1}}{i} \right) \tag{2.11}$$

and because this is an exact potential function, it is isomorphic to a congestion game. Finally, the weighted fractional multicast problem has a Nash equilibrium, with the proof via Kakutani's fixed point theorem [6].

2.3.5 Complexity of Calculating an Optimal Flow

[17] states that it is possible to compute, up to an arbitrarily small error term, an optimal flow for a given instance in time polynomial in the size of the instance and the number of bits of precision required.

2.4 Techniques

In general, much of the work done at the intersection of theoretical computer science and game theory utilises linear programming, convex optimisation and approximation algorithms to achieve the proofs required. However, we also examined a few specific techniques that were particularly useful in the works examined.

2.4.1 Lower Bounding the Price of Anarchy with Simple Networks

In order to find the price of anarchy, the literature, such as [18], proves upper bounds for the price of anarchy, and then shows that this bound cannot be improved upon by way of example, thus proving that the upper bound is tight. [18] proves that such examples can be realised by way of the simplest of networks:

- For a set $\mathscr{C}$ that contains the constant functions, the worst possible value of ρ is obtained by a single-commodity instance on a two-node, two-link network (up to an arbitrarily small additive factor).
- If we relax the constraint that $\mathscr{C}$ must contain the constant functions, the worst possible value of ρ is obtained by a single commodity instance on a network of parallel links.

In fact, the price of anarchy is independent of the complexity of the network topology [16].

2.4.2 Reducing the Size of a Linear Program

In [8], the resulting linear program can be exponential to the graph size, thus destroying the possibility of solving it in polynomial time. The paper utilised two techniques, the second of which was found to be more effective:

- Reformulate an equivalent LP that uses (a polynomial number of) links instead of paths.
- Use Lagrangian relaxation, where, generally, constraints are removed and penalties are placed into the objective function, together with an iterative, subgradient algorithm.

2.4.3 Computing the Price of Anarchy

In [2], the authors noted the general techniques used to find the price of anarchy, in both pure and mixed strategy cases. For the pure case, the authors compared the delay encountered by each agent to the delay it would encounter if it changes to the optimal route, and these delays were combined in some weighted fashion. For mixed strategies, the authors considered the latency of the expected load of each edge, and this was augmented to the expectation of the total frequency.

Chapter 3

Research Plan

This chapter outlines the proposed contribution from this project, the conceptual stages of the project and the proposed timeline in which the stages of the project will be executed.

3.1 Contributions

3.1.1 Reciprocal Congestion Games

As mentioned above, the key contribution of this project is to study reciprocal congestion games, where the cost functions of the Roughgarden traffic model [17] are reciprocated to form functions decreasing in the number of users on an edge. More concretely, we intend to provide the following contributions to the area of reciprocal congestion games:

- Proof of a lower and upper bound on the price of anarchy for the reciprocal congestion game: where this is not possible (for example, where the reciprocal congestion game turns out to have an unbounded price of anarchy), we will attempt to show bounds for restricted classes of networks and cost functions.
- Examination of whether, and what types of, Nash equilibria can be supported by the reciprocal congestion game and its variants: we will examine pure and mixed equilibria, and approximate equilibria, in the modified Roughgarden model, the model with finite splittable and unsplittable flow, and the model with finite edge capacities.
- Examination of whether the ratio of the traffic rates for different commodities affects the price of anarchy.
- Proof of the computational complexity of calculating Nash equilibria in this model: some restricted types of network congestion games possess polynomial time algorithms, and it would be interesting to see if this model possesses this.

3.1.2 Applications to Multicast Routing

As a practical application of the reciprocal congestion game, we consider the application to the selfish multicast routing problem. In particular, we intend to provide the following contributions:

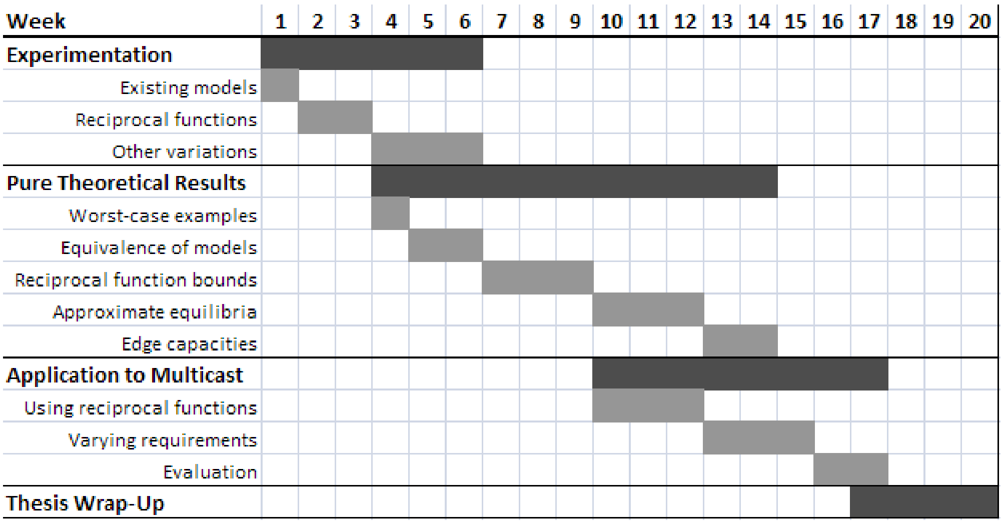

Figure 3.1: Approximate date ranges of the stages to the project: week 1 corresponds to the last week of June 2007, and week 20 is towards the end of October. The three main stages to the project and the thesis wrap-up are designated by a dark grey, while their subcomponents are designated by a lighter grey. The meanings of the subcomponents are discussed in detail in the corresponding text.

- A model of charging multicast users for participation in the transmission that involves reciprocal cost functions with minimal price of anarchy: it is hoped that a generalisation beyond the Shapley value can increase the number of the desirable qualities (mentioned in [5]) that we can satisfy in a model at a time.
- An analysis of the model in terms of computational complexity and network overhead complexity.

3.2 Planning

3.2.1 Stages of the Project

The project involves several stages, which overlap in time, as each successive stage complements the previous. In brief, the stages, together with what each stage involves, are:

- Experimentation and exploration: in this stage, we create models on the computer, and run through optimisation packages and Nash equilibria tools to explore the price of anarchy of the reciprocal congestion game, and variations.
- Pure theoretical results: we attempt to generalise the observations made in the experimentation and exploration stage, and we try to prove theorems about the price of anarchy of the reciprocal congestion game, and variations.
- Application to multicast: we take the theorems about the bounds on the price of anarchy and attempt to derive economic incentives for selfish users to connect to a multicast network that result in a globally optimum result.

Figure 3.1 illustrates the proposed timeline in which the stages of the project will be executed, in terms of the three overarching stages, and the subcomponents of each stage. The three stages

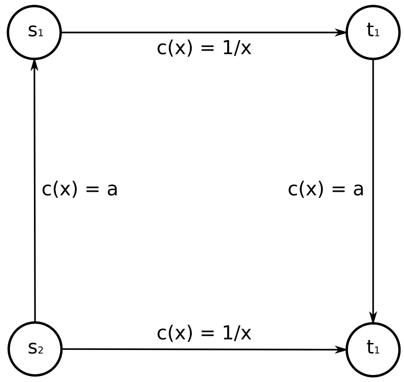

Figure 3.2: A possible network to explore with regards to the price of anarchy. This network has two commodities, and several equilibria. One such equilibria routes traffic for both commodities through the upper edge.

are dealt with in more detail in the following sections.

3.2.2 Experimentation and Exploration

This stage of the project is the first stage, and it is designed to help us gain an intuition into the behaviour of the new class of problems that we will study, and potentially provide some examples that demonstrate cases where the price of anarchy is large.

Exploration of Different Networks

Figure 3.2 is one of the networks that we have come up with to date that demonstrate interesting behaviour (interesting as in, the users' behaviour is non-trivial). We can analyse such networks for their Nash equilibria and their optimal solutions, and thus calculate the price of anarchy; other variations such as finite splittable and unsplittable flows can also be analysed once we have generated a network. This process can be done by hand for small networks or done by using the software packages mentioned below. The first networks to be explored will be small networks created in an ad-hoc manner; this is because, as stated before, small networks capture most of the interesting behaviour to be found.

A more systematic exploration of the available networks will come about by considering classes of networks for which we can calculate the price of anarchy. In particular, we will consider:

- Sparse/dense networks: the sparsity of the network can be modified to explore its effect.
- Statistical distributions over the degrees of the vertices of the graph.
- Statistical distributions over the traffic rates for different commodities: for example, Zipf's Law can be used.

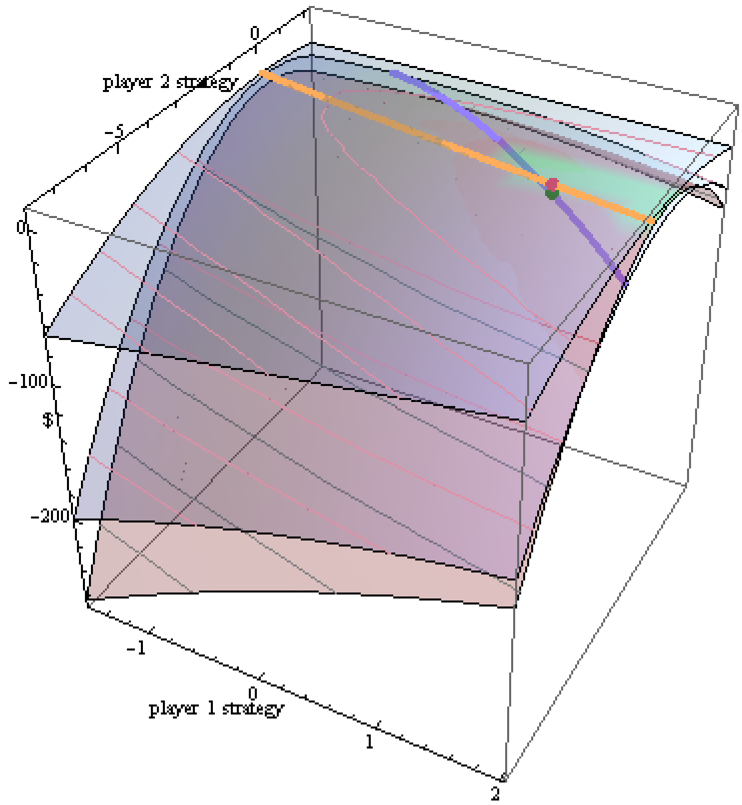

Figure 3.3: A screen shot of a demonstration in Mathematica involving Nash equilibria with continuous strategies. This 3D model can be visualised from a number of angles to facilitate understanding of the features of the best response functions and surfaces.

Software Packages

A number of software packages will be used to assist in the generation and analysis of the networks that we wish to explore:

- AMPL/CPLEX: CPLEX is an optimiser that can accept models expressed in algebraic form; this can be entered in its own language or via AMPL, an alternative modelling language. This does not directly facilitate the analysis of network congestion games; they will have to be expressed as linear or convex programs, in the spirit of [17], and we will need to map the optimal solutions of programs to Nash equilibria via the marginal cost function. The techniques for reducing the size of a linear program, as outlined in Chapter 2, should ensure the feasibility of calculating equilibria on computer.
- GAMUT/Gambit: GAMUT generates games that can be used in Gambit, which is set of software tools that can find Nash equilibria and perform other analyses on games.
- Mathematica: Mathematica is valuable for interactive visual exploration of strategy spaces and their Nash equilibria and optimal solutions (Figure 3.3).

Timeline of Subcomponents

There are a number of subcomponents in this stage (see Figure 3.1):

- Existing models: Models from the literature will be reproduced experimentally with the software packages mentioned above to allow for comparisons when we change the cost functions.
- Reciprocal functions: We will then introduce reciprocal functions in a wide variety of networks, as outlined previously.
- Other variations: Once we have analysed the effect of reciprocal cost functions on with the Roughgarden traffic model, we can then examine the effect of combining the reciprocal cost functions with other variations on the model, including finite splittable and unsplittable flow, and approximate equilibria.

Risks

There are a number of risks in this stage:

- That there do not appear to be any possible generalisations, or all of the networks examined have unbounded price of anarchy: if we cannot find features that allow us to bound the price of anarchy, we will proceed to examining more restricted classes of networks as outlined above.
- The software does not generate networks with the required features: GAMUT, in particular, is documented to generated congestion games only where the congestion is increasing in the number of users. A post-processing stage is possible, but we may have to write our own software to generate these graphs. This is not difficult and will not pose a major threat to the feasibility of the project. There is always the possibility of analysing the networks using the CPLEX optimiser instead of formulating it directly as a game to be solved by Gambit.

3.2.3 Pure Theoretical Results

Timeline of Subcomponents

There are a number of subcomponents in this stage (see Figure 3.1):

- Worst-case examples: based on results from the experimental stage, we will prove that certain networks have a high lower-bound for the price of anarchy. Lower bound examples have often showed the tightness of bounds, as mentioned in Chapter 2.
- Equivalence of models: we will investigate whether the reciprocal congestion game is equivalent to simply treating congestion as a good attribute in the congestion game with normal cost functions. This will be done by comparing the constraints in the corresponding optimisation programs.
- Reciprocal function bounds: we will attempt to prove bounds on the price of anarchy on the reciprocal congestion game. Primarily, we will attempt to use convex programs, which allow us to relate optimisation problems to Nash equilibria.
- Approximate equilibria: [19] contains an example of how to extend the proofs to find a bound for the price of anarchy involving approximate equilibria. This is fairly mechanical, so if the previous subcomponent finds a proof, we should be able to follow the trail to obtain a result here as well.

- Edge capacities: we will add edge capacities to the graph in the spirit of [8]. This will allow us to add economic constraints to influence users, and then remove them later while maintaining an equilibrium as in [8].

Risks

There are a number of risks in this stage:

- The change to reciprocal functions means that we do not automatically have convex functions, in particular, the marginal cost function. Many of the proofs in the literature for upper bounds on the price of anarchy, in particular, [18], rely on the assumption that we have convex programs with certain properties unique to convex programs. We will mitigate this problem by attempting to map the optimisation programs back to convex functions, or if all else fails, to restrict the contribution to a numerical analysis from experimentation.
- We may not find a proof of the upper bound of the price of anarchy in the reciprocal congestion game: if this is not possible, within several weeks of attempting this (simultaneously with the experimentation stage), we will restrict our attention to proof on a restricted network types, as mentioned in the experimental section, or restricted cost function families.
- Edge constraints can also be enforced from first principles by modifying the original optimisation program.

3.2.4 Application to Multicast

Timeline of Subcomponents

There are a number of subcomponents in this stage (see Figure 3.1):

- Using reciprocal functions: we will modify the Shapley value used in previous work to use more general reciprocal functions. We will use computer models to view the affect on the price of anarchy, using synthetically created networks; the trees of limited height method in [8] will be of use to bound the cost of links carrying users that join the network.
- Varying requirements: Time permitting, we will vary combinations of [5] to see if the more general reciprocal functions permit more lax combinations than was found in [5].
- Evaluation: The method in [5] for analysing the computational and network bandwidth complexity of pricing methods may be able to be reapplied to the new model, as it does not rely heavily on properties of the cost functions. The main determination of whether the new model is feasible would be whether we can apply the marginal cost and Shapley value methods to the new model.

Risks

This section is not core to the project, and if we are unable to formulate a solution, it will not be fatal. There are a number of risks in this stage:

- The ability to modify user behaviour via taxes is partly dependent on our ability to add edge capacities, if we follow a similar argument to that in [8].
- The desired properties of multicast pricing methods in [5] may prove to be incompatible for reasons other than the cost functions; not all combinations have been proven to be incompatible in all cases, and only one is cited in [5]. The solution would be to try different combinations.

3.2.5 Thesis Wrap-Up

The number of weeks allocated to this section should be sufficient to allow for a thorough review of the work performed, and the integration of the various threads present in the research into a coherent whole.

Bibliography

[1] R. Ahlswede, N. Cai, S. Li, and R. Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 46(4):1204–1216, 2000.

[2] B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 57–66, New York, NY, USA, 2005. ACM Press.

[3] C. Chekuri, J. Chuzhoy, L. Lewin-Eytan, J. S. Naor, and A. Orda. Non-cooperative multicast and facility location games. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 72–81, New York, NY, USA, 2006. ACM Press.

[4] A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. In *Proceedings of the thirty-sixth annual ACM Symposium on Theory of Computing*, pages 604–612, New York, NY, USA, 2004. ACM Press.

[5] J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63:21–41, 2001.

[6] S. Kakutani. A generalization of brouwer's fixed point theorem. *Duke Mathematical Journal*, 8:457–459, 1941.

[7] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413, 1999.

[8] Z. Li. Min-cost multicast of selfish information flows. In *Proceedings of IEEE INFOCOM 2007*, 2007.

[9] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.

[10] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: budget balance versus efficiency. *Economic Theory*, 18(3):511–533, 2001.

[11] J. E. Nash Jr. Equilibrium points in n-person games. *Proc. Nat. Acad. Sci. US*, 36:48–49, 1950.

[12] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1/2):166–196, 2001.

[13] M. J. Osborne. *A Course in Game Theory*. MIT Press, Cambridge, Massachusetts, 1994.

[14] C. H. Papadimitriou. Algorithms, games, and the internet. In *Proceedings of the thirty-third annual ACM Symposium on Theory of Computing*, pages 749–753, New York, NY, USA, 2001. ACM Press.

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

[16] T. Roughgarden. The price of anarchy is independent of the network topology. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 428–437, New York, NY, USA, 2002. ACM Press.

[17] T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, Cambridge, Massachusetts, 2005.

[18] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.

[19] T. Roughgarden and E. Tardos. Bounding the inefficiency of equilibria in nonatomic congestion games. *Games and Economic Behavior*, 47(2):389–403, 2004.

[20] L. S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.