1 Introduction

1.1 Selfish Routing

What route should you take to work tomorrow? All else being equal, most of us would opt for the one that allows us to wake up at the least barbaric time—that is, most of us would prefer the shortest route available. As any morning commuter knows, the length of time required to travel along a given route depends crucially on the amount of traffic congestion—on the number of *other* commuters who choose interfering routes. In selecting a path to travel from home to work, do you take into account the additional congestion that you cause other commuters to experience? Not likely. Almost certainly you choose your route *selfishly*, aiming to get to work as quickly as possible, without regard to the consequences your choice has for others. Naturally, you also expect your fellow commuters to behave in a similarly egocentric fashion. But what if everyone cooperated by coordinating routes? Is it possible to limit the interference among routes, thereby improving commute times? If so, by how much?

This book studies the loss of social welfare due to *selfish routing*—selfish, uncoordinated behavior in networks. Part II of the book develops techniques for quantifying the worst-possible loss of social welfare from selfish routing, called *the price of anarchy*. Part III uses these techniques to evaluate different approaches to *coping with selfishness*—reducing the price of anarchy with a modest degree of centralized control.

1.2 Two Motivating Examples

This section motivates the questions studied in this book by informally exploring two important examples. These examples are treated rigorously in Chapter 2. Pigou discovered the first example in 1920; Braess discovered the second in 1968.

1.2.1 Pigou's Example

Posit a suburb s and a nearby train station t, connected by two noninterfering highways, and a fixed number of drivers who wish to commute from the suburb s to the train station t at roughly the same time. Suppose the first highway is short but narrow, with the time needed to drive along it increasing sharply with the number of drivers who use it. Suppose the second is wide enough to accommodate all traffic without any crowding, but it takes a long, circuitous route. For concreteness,

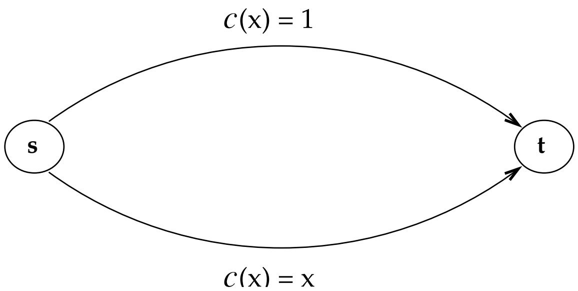
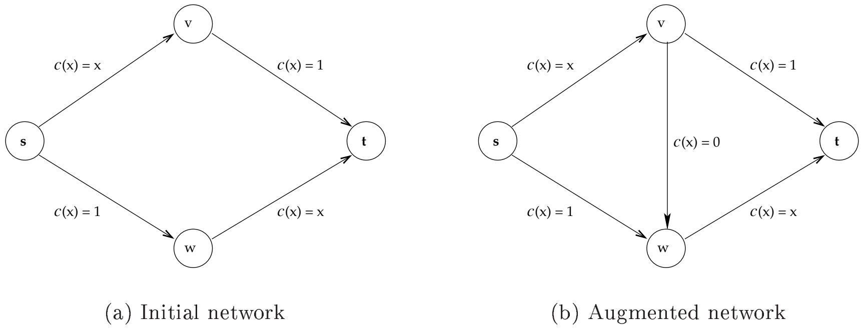


Figure 1.1
Pigou's example. A cost function $c(x)$ describes the travel time experienced by drivers on a road as a function of the fraction x of the overall traffic using that road.

assume that all drivers on the latter highway require 1 hour to drive from s to t, irrespective of the number of other drivers on the road. Further suppose that the time needed to drive using the short narrow highway is equal, in hours, to the fraction of the overall traffic that chooses to use it. Figure 1.1 shows this network pictorially. Call the functions $c(\cdot)$ *cost functions*; in this example they describe the travel time experienced by drivers on a road as a function of the fraction of the traffic that uses the road. The upper edge in Figure 1.1 thus represents the long, wide highway, and the lower edge the short, narrow one.

Assuming that all drivers aim to minimize the driving time from s to t, we have good reason to expect all traffic to follow the lower road and therefore, because of the ensuing congestion, to require one hour to reach the destination t. Indeed, each driver should reason as follows: the lower route is never worse than the upper one, even when it is fully congested, and it is superior whenever some of the other drivers are foolish enough to take the upper route.

Now suppose that, by whatever means, we can choose who drives where. Can the power of centralized control improve over the selfish routing outcome? To see that it can, consider assigning half of the traffic to each of the two routes. The drivers forced onto the long, wide highway experience one hour of travel time, and are thus no worse off than in the previous outcome. On the other hand, drivers allowed to use the short, narrow road now enjoy lighter traffic conditions, and arrive at their destination after a mere 30 minutes. The state of affairs has therefore improved for half of the drivers while no one is worse off. Moreover, the average travel time has dropped from 60 to 45 minutes, a significant improvement. The interested reader might want to ponder whether or not other outcomes are possible in which the



(a) Initial network

(b) Augmented network

Figure 1.2
Braess's Paradox. The addition of an intuitively helpful edge can adversely affect all of the traffic.

average travel time is less than 45 minutes.

Pigou's example demonstrates a well known but important principle:

selfish behavior need not produce a socially optimal outcome.

This observation motivates the work described in Part II, which analyzes the price of anarchy: *how much worse* can a selfish outcome be relative to a socially optimal one? As Part II shows, Pigou's example plays a crucial role in answering this question.

1.2.2 Braess's Paradox

Pigou's example illustrates an important principle: the outcome of selfish behavior need not optimize social welfare. However, it may not be surprising that the result of local optimization by many individuals with conflicting interests does not possess any type of global optimality. The next example, called *Braess's Paradox*, is decidedly less intuitive.

Begin again with a suburb s, a train station t, and a fixed number of drivers who wish to commute from s to t. For the moment, assume two noninterfering routes from s to t, each comprising one long wide road and one short narrow road as shown in Figure 1.2(a). The combined travel time in hours of the two edges in one of these routes is $1+x$, where x is the fraction of the traffic that uses the route. The routes are therefore identical, and traffic should split evenly between them. In this case, all drivers arrive at their destination 90 minutes after their departure from s.

Now, an hour and a half is quite a commute. Suppose that, in an effort to alleviate these unacceptable delays, we harness the finest available road technology to build a very short and very wide highway joining the midpoints of the two existing routes. The new network is shown in Figure 1.2(b), with the new road represented by edge (v, w) with constant cost $c(x) = 0$, independent of the road congestion. How will the drivers react?

We cannot expect the previous traffic pattern to persist in the new network. As in Pigou's example, the travel time along the new route $s \to v \to w \to t$ is never worse than that along the two original paths, and it is strictly less whenever some traffic fails to use it. We therefore expect all drivers to deviate to the new route. Because of the ensuing heavy congestion on the edges (s, v) and (w, t), all of these drivers now experience two hours of travel time when driving from s to t. Braess's Paradox thus shows that the intuitively helpful action of adding a new zero-cost link can negatively impact *all* of the traffic!

Braess's Paradox raises several interesting issues. First, it furnishes a second example of the suboptimality of selfish routing. Indeed, Braess's example demonstrates this principle in a stronger form than does Pigou's: all drivers would *strictly prefer* the coordinated outcome—the original traffic pattern in the network of Figure 1.2(a)—to the one obtained noncooperatively. More importantly, Braess's Paradox shows that the interactions between selfish behavior and the underlying network structure defy intuition and are not easy to predict. When we tackle algorithmic approaches to coping with selfishness in Part III, the counterintuitive moral of Braess's Paradox will be a persistent thorn in our side:

with selfish routing, network improvements can degrade network performance.

1.3 Applications and Caveats

Although this introduction to selfish routing uses the language of road networks, the model has an array of interpretations and applications, some of which this section discusses. (For more, see Section 1.5.) Also, like every mathematical model, this model of selfish routing has made some concessions to the demands of mathematical tractability, at the expense of perfect verisimilitude. We discuss the primary disconnects between selfish routing and reality in Section 1.3.4.

1.3.1 Transportation Networks

Our first interpretation of selfish routing—as road traffic—is consistent with the chronology of its applications. Pigou described his 1920 example in terms of a road network. The model has enjoyed a central position in theoretical transportation research since the 1950's. Hundreds, if not thousands, of papers have studied it and its innumerable extensions. Sections 1.5 and 2.8 survey some of this work.

1.3.2 Computer Networks

More recently, researchers in computer science and electrical engineering discovered two connections between selfish routing and methods of routing information in computer networks, one obvious, the other less so. In order to emphasize the main ideas behind these connections and avoid consideration of a number of details, the following discussion is deliberately kept at a naive level.

The first interpretation of selfish routing for computer networks is for networks that employ so-called *source routing*. Source routing means that if one computer wants to send information to another, then the sender is responsible for selecting a path of data links between the two machines. This task would typically be performed by the computer's software, rather than manually by the actual computer user. In networks with source routing, cost minimization is a natural goal for end users. In this case, the road and computer network interpretations of selfish routing correspond directly.

While the idea of source routing has generated a fair amount of research in the computer networking community, for several reasons it is not common in real networks. Routing is instead usually accomplished in a *distributed* fashion. In distributed routing, a computer selects only a *single* link along which to send information. After the data crosses the link, it is then the next machine's responsibility to see that the information continues toward its destination. The choice of this link can depend on several factors, including the destination of the data and the current network conditions.

A serious problem with distributed routing is that traffic can travel in circles, never arriving at its destination. Ignoring a host of implementation challenges, the following is a solution to this problem. Each computer decides on a positive *length* for the links that emanate from it. Each such link could have a fixed length, or the lengths could be sensitive to the amount of congestion in the network. The length of a path is the sum of the lengths of its individual links, and a *shortest path* between two points is a path with length equal to or less than the length of every other path. Since edge lengths are positive, a shortest path will not cycle back on itself. Routing on shortest paths therefore avoids cycles. Moreover, practical distributed implementations of algorithms that compute shortest paths between all pairs of machines in a network exist, including some that form the basis of popular Internet routing protocols.

Shortest-path routing leaves a key parameter unspecified: the length of each edge. A direct correspondence between selfish routing and shortest-path routing exists if and only if the edge cost functions coincide with the lengths used to define shortest paths. In other words, when an x fraction of the overall network traffic is using an edge with cost function $c(\cdot)$, then the corresponding shortest-path routing algorithm should define the length of the edge as the number $c(x)$. If the cost function c is nonconstant, then this is a congestion-dependent definition of the edge length. In this case, shortest-path routing will route traffic exactly *as if* it is a network with selfish routing (or source routing). This establishes an equivalence between selfish routing and the distributed routing common in real-life computer networks. Section 1.5 gives pointers to rigorous proofs of this equivalence.

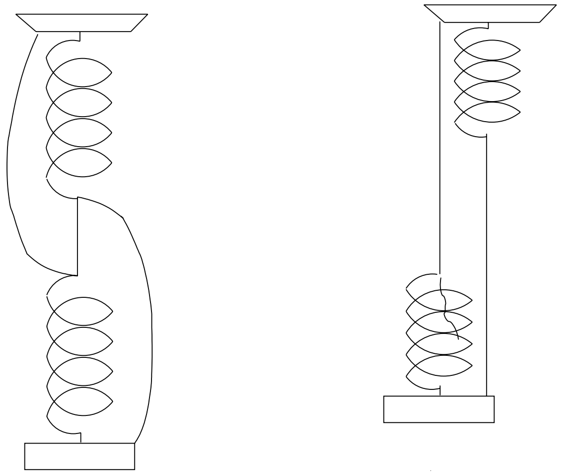
For example, the cost function $c(x)$ of an edge might model the average delay of traffic on the edge, given that an x fraction of the network traffic uses it. Selfish routing with these cost functions models networks in which users pick paths with minimum total delay. Shortest-path routing with these cost functions corresponds to computers defining the length of each outgoing edge as the current average delay experienced by data crossing the edge. The aforementioned equivalence implies that traffic is routed identically in these two different scenarios.

1.3.3 Mechanical and Electrical Networks

Selfish routing also can be relevant in systems that have no explicit notion of traffic whatsoever, as an analogue of Braess's Paradox (Section 1.2.2) in a mechanical network of strings and springs shows.

In the device pictured in Figure 1.3, one end of a spring is attached to a fixed support, and the other end to a string. A second identical spring is hung from the free end of the string and carries a heavy weight. Finally, strings are connected, with some slack, from the support to the upper end of the second spring and from the lower end of the first spring to the weight. Assuming that the springs are ideally elastic, the stretched length of a spring is a linear function of the force applied to it. We can therefore view the network of strings and springs as a traffic network, where force corresponds to traffic and physical distance corresponds to cost.

With a suitable choice of string and spring lengths and spring constants, the equilibrium position of this mechanical network is described by Figure 1.3(a). Perhaps unbelievably, severing the taut string causes the weight to *rise*, as shown in Figure 1.3(b)! An explanation for this curiosity follows. Initially, the two springs are connected in series, and each bears the full weight and is stretched out to great length. After cutting the taut string, the two springs are only connected in parallel. Each spring then carries only half of the weight, and accordingly is stretched to only half of its previous length. The rise in the weight is the same as the improvement



(a) Before

(b) After

Figure 1.3
Strings and springs. Severing a taut string lifts a heavy weight.

in the selfish outcome obtained by deleting the zero-cost edge of Figure 1.2(b) to obtain the network of Figure 1.2(a). Because such systems of strings and springs are essentially the same as networks with selfish routing, the bounds on the price of anarchy that Chapter 3 describes also limit the largest-possible magnitude of this counterintuitive effect.

Similarly, removing a conducting link from an electrical network can increase its conductivity. Electrical networks are again the same as networks with selfish routing, so bounds on the price of anarchy translate to limits on this increase of conductivity.

1.3.4 Caveats

This section has demonstrated that selfish routing is a versatile model that captures key features of a diverse collection of applications. The model does, however, possess some weaknesses, especially in the context of routing in Internet-like computer networks. Two of these follow, along with a critique of the price of anarchy. While this is not an exhaustive list of the model's flaws, these are arguably the most fundamental. Many other assumptions made by the model can be removed, as

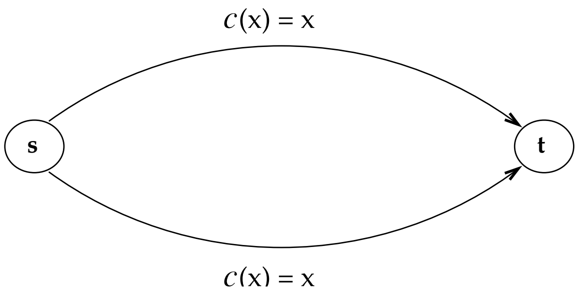


Figure 1.4
A possibly unstable network.

Chapter 4 shows.

The first criticism of selfish routing applies to both the road and computer network interpretations of the model: the model is *static*, while the world is *dynamic*. When we prove bounds on the price of anarchy, we will assume that the network has reached an "equilibrium." We already have an informal sense of what this means from the examples in Section 1.2; Section 2.2 defines the notion formally. Conceptually, the hope is that traffic will experiment over time and reach an equilibrium, but it is not clear that this will always occur, especially in networks where parameters such as the traffic rate are changing rapidly over time. For a contrived example, replace the long, wide highway in Pigou's example with a second short, narrow one (Figure 1.4). Imagine a computer at s that routes traffic using shortest-path routing with the cost functions c. Perhaps at first the computer routes all of the traffic on the upper route. Then, finding that the upper edge has cost 1 and the lower edge cost 0, the computer reconsiders and routes all of the traffic on the lower edge. This undesirable oscillation can continue unabated.

In defense of studying equilibria, a variety of reasonably weak conditions are known to be sufficient for a network with selfish routing to settle into an equilibrium. In the contrived example above, the obvious solution is to restrict the amount of traffic that can be rerouted at each time step.

Second, shortest-path routing in computer networks, and in the Internet in particular, poses a particular challenge for the model. A network like the Internet is volatile. Its traffic patterns can change quickly and dramatically, and data links and machines are constantly failing. The assumption of a static model is therefore particularly suspect in such networks. Indeed, early versions of the Internet used variants of delay-based distributed routing and suffered from unstable behavior.

Moreover, routing in the Internet today is often not sensitive to congestion. In addition to its relative stability, this naive routing is common because economic factors tend to overwhelm performance-based ones: cheap links are used even when they are very congested. Of course, there is little hope of proving anything about performance-ignorant routing. Indeed, recent experiments have demonstrated that current routing in the Internet is highly inefficient. Because of this, congestion-sensitive routing has been making a bit of a comeback, at least within the research community. If widely adopted, this would improve the accuracy of the correspondence between selfish routing and distributed routing in large, real-life networks. See Section 1.5 for references on these developments.

As a final critique, the price of anarchy—defined in Section 2.3 as the worst-possible ratio between the average travel time of a selfish solution and the smallest achievable average travel time—is by definition a worst-case measure. Worst-case analysis has long inspired heated debate. Worst-case bounds are of course compelling when they exist, but worst-case analysis can focus undue attention on contrived bad examples—akin to ignoring a lush forest for the sake of a few dead trees. On the other hand, alternatives to worst-case analysis usually must define what a "typical case" or an "average-case analysis" means, and no such definition is without its own controversy. Also, these alternatives are often less mathematically tractable than their worst-case counterparts.

Most of this book adopts the worst-case approach. Fortunately, interesting worst-case bounds can usually be established for the price of anarchy. Moreover, Part II demonstrates that worst-case examples are often similar to networks that can arise in practice. Nevertheless, the pursuit of alternatives to worst-case analysis remains an important and largely unexplored research direction.

1.4 How to Read this Book

1.4.1 Prerequisites

This book has few prerequisites, other than mathematical maturity. On occasion it assumes a bit more, as follows. A few proofs assume that the reader remembers some basic calculus and analysis, mostly to use derivatives to approximate a function, and to use the fact that a continuous real-valued function defined on closed and bounded subset of Euclidean space attains its minimum. The reader with no exposure to mathematical programming should skip the proofs in Sections 2.4 and 2.6 and that of Lemma 4.3.6. The reader is assumed to be familiar with the theory of NP-completeness only in Sections 5.3 and 6.6, although the language of

Table 1.1
Prerequisites needed for particular sections of the book.

Prerequisite	Needed for parts of...
Calculus/Analysis	Sections 2.4, 2.6, 3.2, 3.5 and 4.3
Convex Programming	Sections 2.4, 2.6, and 4.3
NP-Completeness	Sections 5.3 and 6.6

complexity theory also occurs in a few other places. To navigate these, the reader unfamiliar with computational complexity can simply translate "polynomial-time" as "computationally efficient," "NP-complete" and "NP-hard" as "computationally intractable," and take $P \neq NP$ as an axiom. See Table 1.1 for a summary of these prerequisites.

Prior exposure to the language of graphs, as provided by an undergraduate course in algorithms or in combinatorics, will also be very useful. Experience with network flow or combinatorial optimization is ideal. The presentation here is self-contained, but it will probably seem a bit terse to readers without this background. Finally, while the book sometimes uses the language of game theory, no prior knowledge of the field is required. The chapter notes also discuss some of the connections between the work described in this book and classical game theory.

1.4.2 Dependencies

Chapter 2 is a prerequisite for all that follows, though most of its sections are required only for a subset of the rest of the book. This breakdown is discussed in detail in the chapter's introduction. Chapters 3, 5, and 6 can be read independently of each other, although Chapter 5 uses some of the results from Chapter 3. Chapter 4 is meant to be read after Chapter 3.

1.5 Notes

Section 1.1

The term "selfish routing" is originally due to Roughgarden and Tardos [347], though the mathematical model that it refers to is much older. This model was discussed qualitatively by Pigou [320] and Knight [227] in the 1920s. In the 1950s, Wardrop [394] and Beckmann, McGuire, and Winsten [40] formalized the model, and selfish routing has been intensely studied ever since. Chapter 2 discusses this history in much greater detail. There have also been innumerable other applications of game theory to networks over the past several decades. See the end of Section 3.7; the papers by Shenker [360], Altman et al. [10], and Linial [258]; and the references therein for many examples.

The concept of the price of anarchy originated in Koutsoupias and Papadimitriou [239], where it was called the *coordination ratio*. Section 3.7 discusses in depth the model studied in [239]. The phrase "price of anarchy" was coined by Papadimitriou [303]. There were several precursors to this concept, however, as Section 2.8 discusses.

"Coping with selfishness" is meant to parallel the expression "coping with NP-completeness," which was popularized by Garey and Johnson [172] and refers to methods for evading the (presumed) worst-case computational intractability of important NP-complete problems. Part III of this book describes only two of the many known ways to reduce the price of anarchy; for example, the vast and important topic of economic incentives, such as taxes and subsidies, will be touched on only briefly in Sections 5.3 and 6.7.

Section 1.2

Pigou's example and Braess's Paradox first appeared in Pigou [320] and Braess [64], respectively. Murchland [285] was the first to describe Braess's Paradox in English. The quantitative details in Section 1.2, such as the network cost functions, are somewhat different than in Pigou's and Braess's original formulations. They are taken from Roughgarden and Tardos [348] and L. Schulman (personal communication, October 1999), respectively. Chapter 5 studies Braess's Paradox in depth.

The first moral of Section 1.2, that selfish behavior need not yield a socially optimal outcome, is arguably as old as economics itself. In the language of economics, this moral states that a selfish outcome can be *Pareto inefficient*—there can be a different outcome in which someone is better off while no one is worse off. Perhaps the most canonical example in game theory of the Pareto inefficiency of selfish behavior is the *Prisoner's Dilemma*. In the Prisoner's Dilemma, two prisoners have been captured and are to be interrogated separately by the authorities. Each prisoner has to plead guilty or not guilty to the charges and is aware of the following possible outcomes. If both prisoners plead not guilty, the authorities lack sufficient evidence to convict them, and they will both go free. If both plead guilty, then both receive moderate jail terms. If exactly one of them pleads guilty, then the confessor is set free and given a reward for the information, while the other prisoner is given a draconian jail sentence. If the prisoners could coordinate, they would both plead not guilty. In the absence of cooperation, however, the incentives are clear: each prisoner is strictly better off by confessing, no matter what the other prisoner says. We therefore expect selfish behavior to result in both prisoners confessing and serving time. As in Braess's Paradox, everyone suffers in the selfish outcome, relative to what can be achieved with cooperation. For more on the Prisoner's Dilemma and its history, see Raiffa [325] and the book by Rapoport and Chammah [326]. For further discussion of the Pareto inefficiency of selfish behavior, see Dubey [128] or Cohen [81].

Analogues of the second moral of Section 1.2 are also known in many different contexts; some of these are surveyed in Section 5.4.

Section 1.3

Selfish routing was originally introduced to model transportation networks, and Chapter 2 discusses this history in detail. Its application to computer networks, with both source and distributed routing, is somewhat more recent. Cantor and Gerla [72], Gallager [170], and Stern [375] are good early references on the topic. Bertsekas and Tsitsiklis [51]—especially Sections 3.5, 5.6, and 7.6—is a nice textbook treatment. Both [72] and [51], and more recently Friedman [164], give details on the equivalence between selfish and shortest-path routing discussed in Section 1.3.

Most networking textbooks contain far more details about the nuts and bolts of implementing a distributed routing protocol than are included in Section 1.3. Examples include Bertsekas and Gallagher [50], Keshav [224], Peterson and Davie [318], and Walrand [393].

The mechanical and electrical network examples of Section 1.3.3 were first given by Cohen and Horowitz [82]. Other connections between traffic equilibria and these physical networks were discussed earlier by Duffin [129] and Enke [134]. Penchina and Penchina [314] offer advice on realizing the strings and springs system of Figure 1.3.

Just as the applications of Sections 1.3.1–1.3.3 are intended to demonstrate that selfish routing is a useful and flexible mathematical model, the caveats of Section 1.3.4 are meant as a caution against overzealous interpretations of the results of this book. For example, despite the connotations of a recent *New York Times* article [26], there is no easy way to translate the theoretical work of this book into a better Internet. Nonetheless, the techniques described in this book should find use as a tool to analyze congestion-sensitive routing in a variety of network applications.

The first caveat of Section 1.3.4 is that the selfish routing model studied in this book assumes convergence to an equilibrium; the example shown in Figure 1.4 is essentially due to Friedman (see [26]). This issue is reasonably well understood for transportation networks. First, there are numerous more general models of selfish routing that explicitly account for dynamics; some of these are surveyed in Section 2.8. Second, convergence to the static equilibrium in such networks has been verified both experimentally and theoretically. See [146, 156, 165, 251, 262, 291, 368, 405], for example, for further details.

Khanna and Zinky [226] and the references therein describe how delay-based distributed routing was implemented in the early Internet, and discuss how these implementations can fail to converge to an equilibrium. See also Keshav [224, §11.7]. On the other hand, careful implementations can be proven to converge in networks that are not too volatile. See, for example, Bertsekas and Tsitsiklis [51] and the related research papers by Tsitsiklis and Bertsekas [385] and Tsai, Tsitsiklis, and Bertsekas [384]. Other ways to improve the stability of congestion-sensitive shortest-path routing are given by, among others, Khanna and Zinky [226] and Chen, Druschel, and Subramanian [76].

Savage et al. [352] describe experiments that demonstrate the inefficiency of current (congestion-insensitive) Internet routing and discuss some possible alternatives. The potential benefits of congestion-sensitive routing in the Internet are also mentioned in many other papers, including in [76, 226].

Worst-case analysis has long been central to theoretical computer science, dating back to an early obsession with the worst-case running time of computer algorithms (see e.g. [4]). For further discussion of its merits and drawbacks see, for example, Aho, Hopcroft, and Ullman [4], Ahuja, Magnanti, and Orlin [5], or Kozen [240]. Chapters 5 and 6 of this book consider approximation algorithms, which exemplify another type of worst-case analysis.

Prerequisites

The following are some favorite references for the topics that Section 1.4.1 discusses. The few facts from calculus and analysis used here are covered in Mardsen and Hoffman [268], Rudin [350], and Spivak [371]. Hillier and Lieberman [193] and Peressini, Sullivan, and Uhl [316] provide elementary introductions to mathematical programming. Garey and Johnson [172] and Papadimitriou [302] are good references for NP-completeness. Ahuja, Magnanti, and Orlin [5], Cook et al. [90], Papadimitriou and Steiglitz [306], and Tarjan [380] are among the many excellent texts on combinatorial optimization and network flow. Lastly, standard introductions to game theory include Fudenberg and Tirole [166], Mas-Colell, Whinston, and Green [270], Osbourne and Rubinstein [300], and Owen [301]. Also, Straffin [377] is good for a quicker, less technical introduction to game theory.