CS364A: Algorithmic Game Theory
Lecture #11: Selfish Routing and the Price of Anarchy*

Tim Roughgarden†

October 28, 2013

1 Quantifying the Inefficiency of Equilibria

With this lecture we begin the second part of the course. In many settings, you do not have the luxury of designing a game from scratch. When do games "in the wild" have near-optimal equilibria?

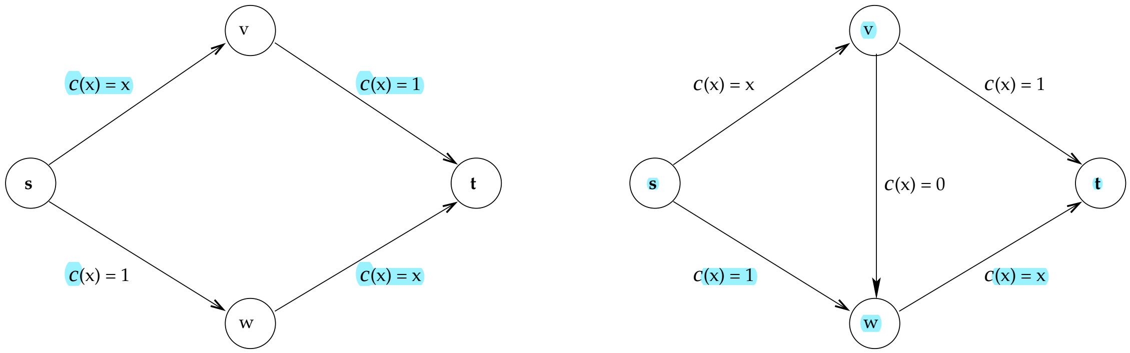
Unlike all of our carefully crafted DSIC mechanisms, games in the wild generally have no dominant strategies. As a consequence, predicting the game's outcome requires stronger assumptions about how players behave. There's also no reason to expect optimal outcomes in games that we didn't design. Even for approximation results, we can only hope for success in fairly specific — though hopefully interesting and relevant — application domains.

In all of our mechanism design applications, strategic players held private information. In this part of the course, we'll focus only on full-information games, where the payoffs of all players are common knowledge. For example, when we consider network traffic, we'll assume that everyone prefers shorter routes to longer routes. By contrast, in a single-item auction with private information, one bidder has no idea whether a different bidder would prefer to win at some price p or would prefer to lose (at price 0). In later advanced lectures, we'll develop techniques for quantifying the inefficiency of equilibria also in games with incomplete information, such as auctions. The research agendas of mechanism design and quantifying inefficiency in games are not at odds, and have usefully intertwined in the past few years.

The foundations of mechanism design were laid decades ago in the economics literature; while the computer science community has helped advance the basic theory, its bigger contributions have been through novel questions, mathematical techniques, and applications. By contrast, all of the research from this part of the course will be from the last 15 (and mostly 5-10) years, and the origins of the area are in theoretical computer science [3, 6].

---

†Department of Computer Science, Stanford University, 462 Gates Building, 353 Serra Mall, Stanford, CA 94305. Email: `tim@cs.stanford.edu`.



(a) Initial network

(b) Augmented network

Figure 1: Braess's Paradox. After the addition of the (v, w) edge, the price of anarchy is 4/3.

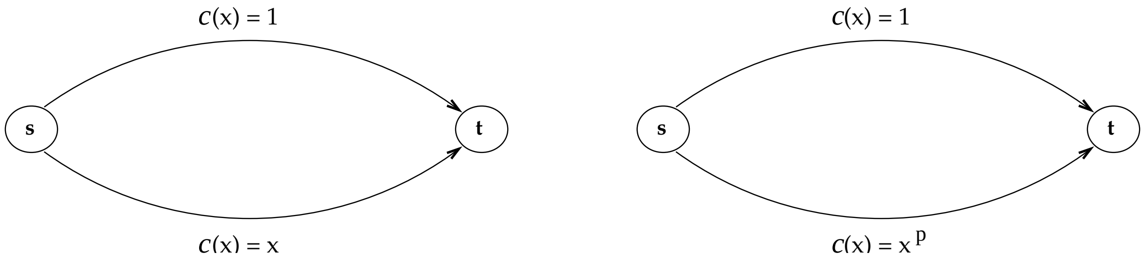
2 Selfish Routing: Examples

2.1 Braess's Paradox

Recall Braess's Paradox from Lecture 1 (Figure 1) [1]. One unit of traffic (e.g., rush-hour drivers) leave from s to t at the same time. Each edge is labeled with a cost function, which describes the travel time of all traffic on the edge, as a function of the fraction x of traffic that uses it. In the network in Figure 1(a), by symmetry, in equilibrium half of the traffic uses each route, and the travel time experienced by all traffic is $\frac{3}{2}$. After installing a teleportation device allowing drivers to travel instantly from v to w (Figure 1(b)), however, it is a dominant strategy for every driver to take the new route $s \to v \to w \to t$. The common travel time in this new equilibrium is 2. The minimum-possible travel time in the new network is $\frac{3}{2}$ — there is no profitable way to use the teleporter. If we define the *price of anarchy (POA)* to be the ratio between the travel time in an equilibrium and the minimum-possible average travel time, then the price of anarchy in the Braess network (Figure 1(b)) is $\frac{4}{3}$.

2.2 Pigou's Example

There is an even simpler selfish routing network in which the POA is $\frac{4}{3}$, first discussed in 1920 by Pigou [4]. In Pigou's example (Figure 2(a)), every driver has a dominant strategy to take the lower link — even when congested with all of the traffic, it is no worse than the alternative. Thus, in equilibrium all drivers use the lower edge and experience travel time 1. Can we do better? Sure — any other solution is better! An altruistic dictator would minimize the average travel time by splitting the traffic equally between the two links. This results in an average travel time of $\frac{3}{4}$, showing that the POA in Pigou's example is $\frac{4}{3}$.



(a) Pigou's example (b) A nonlinear variant

Figure 2: Pigou's example and a nonlinear variant. The cost function $c(x)$ describes the cost incurred by users of an edge, as a function of the amount of traffic routed on the edge.

2.3 Nonlinear Pigou's Example

The POA is $\frac{4}{3}$ in both Braess's Paradox and Pigou's example — not so bad for completely unregulated behavior. The story is not so rosy in all networks, however. In the nonlinear Pigou's example (Figure 2(b)), we replace the previous cost function $c(x) = x$ of the lower edge with the function $c(x) = x^p$, with p large. The lower edge remains a dominant strategy, and the equilibrium travel time remains 1. What's changed is that the optimal solution is now much better. If we again split the traffic equally between the two links, then the average travel time tends to $\frac{1}{2}$ as $p \to \infty$ — traffic on the bottom edge gets to t nearly instantaneously. We can do even better by routing $(1-\epsilon)$ traffic on the bottom link, where ϵ tends to 0 as p tends to infinity — almost all of the traffic gets to t with travel time $(1-\epsilon)^p$, which is close to 0 when p is sufficiently large, and the ϵ fraction of martyrs on the upper edge contribute little to the average travel time. We conclude that the POA in the nonlinear Pigou's example is unbounded as $p \to \infty$.

3 Main Result: Statement and Interpretation

The POA of selfish routing can be large (Section 2.3) or small (Section 2.1 and 2.2). The goal of this lecture is to provide a thorough understanding of when the POA of selfish routing is close to 1. Looking at our three examples thus far, we see that "highly nonlinear" cost functions can prevent a selfish routing network from having a POA close to 1, while our two examples with linear cost functions have a small POA. The coolest statement that might be true is that highly nonlinear cost functions are the *only* obstacle to a small POA — that every network with not-too-nonlinear cost functions, no matter how complex, has POA close to 1. The main result of this lecture formulates and proves this conjecture.

The model we study is the following. There is a directed graph $G = (V, E)$, with a source vertex s and a sink vertex t. There are r units of traffic (or *flow*) destined for t from s.[1] We

[1] To minimize notation, we prove the main result only for single-commodity networks, where there is one source and one sink. The main result and its proof extend easily to networks with multiple sources and

treat G as a flow network, in the sense of the maximum- and minimum-cost flow problems.

Each edge e of the network has a *cost function*, describing travel time (per unit of traffic) as a function of the amount of traffic on the edge. Edges do not have explicit capacities. For our selfish routing lectures, we always assume that every cost function is non-negative, continuous, and nondecreasing. These are very mild assumptions in most relevant applications, like road or communication network traffic.

We first state an informal version of this lecture's main result and explain how to interpret and use it. We give a formal statement at the end of this section and a proof in Section 5. Importantly, the theorem is parameterized by a set $\mathcal{C}$ of permissible cost functions. This reflects our intuition that the POA of selfish routing seems to depend on the "degree of nonlinearity" of the network's cost functions. The result is already interesting for simple classes $\mathcal{C}$ of cost functions, such as the set $\{c(x) = ax + b \,:\, a, b \geq 0\}$ of affine functions with nonnegative coefficients.

Theorem 3.1 (Tight POA Bounds for Selfish Routing (Informal) [5]) *Among all networks with cost functions in a set $\mathcal{C}$, the largest POA is achieved in a Pigou-like network.*

The point of Theorem 3.1 is that worst-case examples are always simple. The principal culprit for inefficiency in selfish routing is nonlinear cost functions, not complex network structure.

For a particular cost function class $\mathcal{C}$ of interest, Theorem 3.1 reduces the problem of computing the worst-case POA to a back-of-the-envelope calculation. Without Theorem 3.1, one would effectively have to search through all networks with cost functions in $\mathcal{C}$ to find the one with the largest POA. Theorem 3.1 guarantees that the much simpler search through Pigou-like examples is sufficient.

For example, when $\mathcal{C}$ is the set of affine cost functions, Theorem 3.1 implies that Pigou's example (Section 2.2) maximizes the POA. Thus, the POA is always at most $\frac{4}{3}$ in selfish routing networks with affine cost functions. When $\mathcal{C}$ is the set of polynomials with nonnegative coefficients and degree at most d, Theorem 3.1 implies that the worst example is the nonlinear Pigou's example (Section 2.3). It is straightforward to compute the POA in this worst-case example, and the result of this computation is an upper bound on the POA of every selfish routing network with such cost functions. See Table 1 for some examples, which makes clear the point that the POA of selfish routing is large only in networks with "highly nonlinear" cost functions. For example, quartic functions have been proposed as a reasonable model of road traffic in some situations [7]. The worst-case POA with respect to such function is slightly above 2. We discuss cost functions germane to communication networks in the next lecture.

We next work toward a formal version of Theorem 3.1. This requires formalizing the "Pigou-like networks" for a class $\mathcal{C}$ of cost functions. We then define a lower bound on the POA based solely on these trivial instances. The formal version of Theorem 3.3 states a matching upper bound on the POA of every selfish routing network with cost functions in $\mathcal{C}$.

sinks; see the Exercises.

Table 1: The worst-case POA in selfish routing networks with cost functions that are polynomials with nonnegative coefficients and degree at most d.

Description	Typical Representative	Price of Anarchy
Linear	$ax+b$	$4/3$
Quadratic	ax^2+bx+c	$\frac{3\sqrt{3}}{3\sqrt{3}-2} \approx 1.6$
Cubic	ax^3+bx^2+cx+d	$\frac{4\sqrt[3]{4}}{4\sqrt[3]{4}-3} \approx 1.9$
Quartic	$ax^4+\cdots$	$\frac{5\sqrt[4]{5}}{5\sqrt[4]{5}-4} \approx 2.2$
Polynomials of degree $\leq d$	$\sum_{i=0}^{d} a_i x^i$	$\frac{(d+1)\sqrt[d]{d+1}}{(d+1)\sqrt[d]{d+1}-d} \approx \frac{d}{\ln d}$

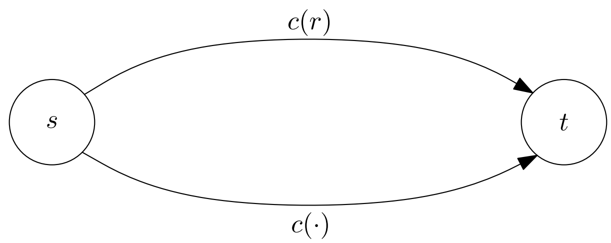


Figure 3: Pigou-like network.

Definition 3.2 (Pigou-like Network) A *Pigou-like network* has:

- Two vertices, s and t.
- Two edges from s to t.
- A traffic rate $r > 0$.
- A cost function $c(\cdot)$ on the first edge.
- The cost function everywhere equal to $c(r)$ on the second edge.

See also Figure 3. Note that there are two free parameters in the description of a Pigou-like network, the traffic rate r and the cost function $c(\cdot)$ of the first edge.

The POA of a Pigou-like network is easy to understand. By construction, the lower edge is a dominant strategy for all traffic — it is no less attractive than the alternative (with constant cost $c(r)$), even when it is fully congested. Thus, in the equilibrium all traffic travels on the lower edge, and the total travel time is $r \cdot c(r)$ — the amount of traffic times the per-unit travel time experienced by all of the traffic. We can write the minimum-possible total travel time as

$$\inf_{0 \leq x \leq r} \{x \cdot c(x) + (r-x) \cdot c(r)\}, \tag{1}$$

where x is the amount of traffic routed on the lower edge. It will be convenient later to let x range over all nonnegative reals. Since cost functions are nondecreasing, this larger range does not change the quantity in (1) — there is always an optimal choice of x in $[0, r]$. Thus, the POA in a Pigou-like network with traffic rate $r > 0$ and upper edge cost function $c(\cdot)$ is

$$\sup_{x \geq 0} \left\{ \frac{r \cdot c(r)}{x \cdot c(x) + (r - x) \cdot c(r)} \right\}.$$

Let $\mathcal{C}$ be an arbitrary set of nonnegative, continuous, and nondecreasing cost functions. Define the *Pigou bound* $\alpha(\mathcal{C})$ as the worst POA in a Pigou-like network with cost functions in $\mathcal{C}$. Formally,

$$\alpha(\mathcal{C}) := \sup_{c \in \mathcal{C}} \sup_{r \geq 0} \sup_{x \geq 0} \left\{ \frac{r \cdot c(r)}{x \cdot c(x) + (r - x) \cdot c(r)} \right\}. \tag{2}$$

The first two suprema simply search over all choices of the two free parameters $c \in \mathcal{C}$ and $r \geq 0$ in a Pigou-like network; the third computes the best-possible routing in the chosen Pigou-like network.

The Pigou bound can be evaluated explicitly for many sets $\mathcal{C}$ of interest. For example, if $\mathcal{C}$ is the set of affine (or even concave) functions, then $\alpha(\mathcal{C}) = \frac{4}{3}$; see the Problems. The expressions in Table 1 are precisely the Pigou bounds for sets of polynomials with nonnegative coefficients and bounded degree. The Pigou bound is achieved for these sets of cost functions by the nonlinear Pigou example (Section 2.3).

By construction, if $\mathcal{C}$ contains all of the constant functions, then $\alpha(\mathcal{C})$ is a lower bound on the worst-case POA of selfish routing networks with cost functions in $\mathcal{C}$ — even for just the Pigou-like networks in this family.[2] The formal statement of Theorem 3.1 is that the Pigou bound $\alpha(\mathcal{C})$ is an upper bound on the POA of *every* selfish routing network with cost functions in $\mathcal{C}$, whether Pigou-like or not.

Theorem 3.3 (Tight POA Bounds for Selfish Routing (Formal) [5, 2]) *For every set $\mathcal{C}$ of cost functions and every selfish routing network with cost functions in $\mathcal{C}$, the POA is at most $\alpha(\mathcal{C})$.*

4 Technical Preliminaries

Before proving Theorem 3.3, we review some flow network preliminaries. While notions like flow and equilibria are easy to define in Pigou-like networks, defining them in general networks requires a little care.

Let $G = (V, E)$ be a selfish routing network, with r units of traffic traveling from s to t. Let $\mathcal{P}$ denote the set of s-t paths of G, which we assume is non-empty. A *flow* describes how traffic is split over the s-t paths — it is a non-negative vector $\{f_P\}_{P \in \mathcal{P}}$ with $\sum_{P \in \mathcal{P}} f_P = r$. For example, in Figure 4, half of the traffic takes the zig-zag path $s \to v \to w \to t$, while the other half is split equally between the two two-hop paths.

[2] As long as $\mathcal{C}$ contains at least one function c with $c(0) > 0$, the Pigou bound is a lower bound on the POA of selfish routing networks with cost functions in $\mathcal{C}$. The reason is that Pigou-like networks can be simulated by slightly more complex networks under this weaker assumption; see the Exercises for details.

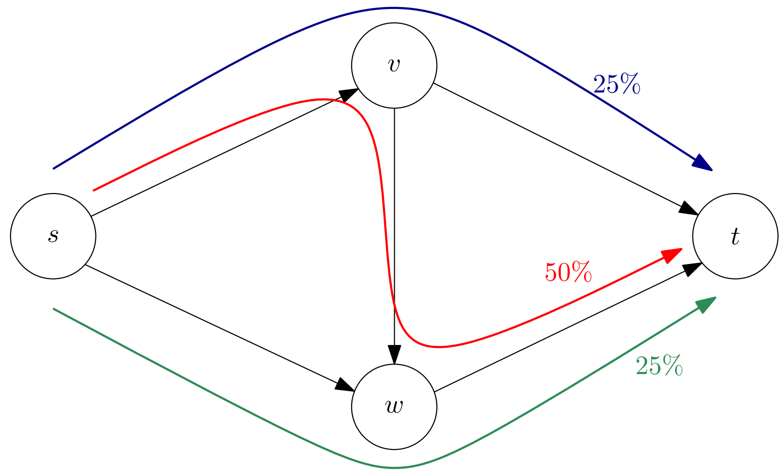


Figure 4: Example flow.

For an edge $e \in E$ and a flow f, we write $f_e = \sum_{P \in \mathcal{P} \,:\, e \in P} f_p$ for the amount of flow that uses a path that includes e. For example, in Figure 4, $f_{(s,v)} = f_{(w,t)} = \frac{3}{4}$, $f_{(s,w)} = f_{(v,t)} = \frac{1}{4}$, and $f_{(v,w)} = \frac{1}{2}$.

A flow is an *equilibrium* if and only if traffic travels only on shortest s-t paths. Formally, $f_{\widehat{P}} > 0$ only if

$$\widehat{P} \in \underset{P \in \mathcal{P}}{\operatorname{argmin}} \left\{ \underbrace{\sum_{e \in P} c_e(f_e)}_{:= c_P(f)} \right\}.$$

Note that "shortest" is defined using the travel times $\{c_e(f_e)\}$ with respect to the flow f. For example, the flow in Figure 4 is not an equilibrium because the only shortest path is the zig-zag path, and some of the flow doesn't use it.

A non-obvious fact is that, in every selfish routing network, there is at least one equilibrium flow. We'll sketch a proof of this later, in Lecture 13.[3]

Our objective function is the total travel time incurred by traffic, and this is denoted by $C(f)$ for a flow f. We sometimes call the total travel time the *cost* of a flow. This objective function can be computed in two different ways, and both ways are useful. First, we can tally travel time path-by-path:

$$C(f) = \sum_{P \in \mathcal{P}} f_P \cdot c_P(f).$$

Alternatively, we can tally it edge-by-edge:

$$C(f) = \sum_{e \in E} f_e \cdot c_e(f_e).$$

[3]Our assumption that cost functions are continuous is important for this existence result.

Recalling that $c_P(f) = \sum_{e \in P} c_e(f_e)$ and $f_e = \sum_{P \in \mathcal{P} : e \in P} f_P$ by definition, a simple reversal of sums formally shows the equivalence of the two expressions above.

A second non-trivial fact about equilibrium flows is that all such flows have the same cost; we'll sketch a proof of this in a later lecture. Thus, it makes sense to define the *price of anarchy (POA)* of a selfish routing network as the ratio between the cost of an equilibrium flow and the cost of an optimal (minimum-cost) flow.

5 Main Result: Proof

We now prove Theorem 3.3. Fix a network G with cost functions in $\mathcal{C}$. Let f and f^* denote equilibrium and optimal (minimum-cost) flows, respectively. The proof has two parts.

The first part of the proof shows that if we "freeze" all edge costs at their equilibrium values $c_e(f_e)$, then the equilibrium flow f is in fact optimal. This should be intuitive since an equilibrium flow routes all traffic on shortest paths with respect to the edge travel times it induces. This does not contradict the sub-optimality of equilibrium flows, as other flows generally induce different edge costs.

Formally, since f is an equilibrium flow, if $f_{\widehat{P}} > 0$, then $c_{\widehat{P}}(f) \leq c_P(f)$ for all $P \in \mathcal{P}$. In particular, all paths $\widehat{P}$ used by the equilibrium flow have a common cost $c_{\widehat{P}}(f)$, call it L. Moreover, $c_P(f) \geq L$ for every path $P \in \mathcal{P}$. Thus,

$$\sum_{P \in \mathcal{P}} \underbrace{f_P}_{\text{sums to } r} \cdot \underbrace{c_P(f)}_{= L \text{ if } f_P > 0} = r \cdot L \tag{3}$$

while

$$\sum_{P \in \mathcal{P}} \underbrace{f^*_P}_{\text{sums to } r} \cdot \underbrace{c_P(f)}_{\geq L} \geq r \cdot L. \tag{4}$$

Rewriting the left-hand sides of (3) and (4) as sums over edges and subtracting (3) from (4) yields

$$\sum_{e \in E} (f^*_e - f_e) c_e(f_e) \geq 0. \tag{5}$$

The "variational inequality" in (5) is stating something very intuitive: since the equilibrium flow f routes all traffic on shortest paths, no other flow f^* can be better if we keep all edge costs fixed at $\{c_e(f_e)\}_{e \in E}$.

The second part of the proof quantifies the extent to which the optimal flow f^* can be better than f. The rough idea is to show that, edge by edge, the gap in costs between f and f^* is no worse than the Pigou bound. This statement only holds up to an error term for each edge, but we can control the sum of the error terms using the inequality (5) from the first part of the proof.

Formally, for each edge $e \in E$, instantiate the right-hand side of the Pigou bound (2) using c_e for c, f_e for r, and f^*_e for x. Since $\alpha(\mathcal{C})$ is a supremum over all possible choices of

c, r, and x, we have

$$\alpha(\mathcal{C}) \geq \frac{f_e \cdot c_e(f_e)}{f_e^* \cdot c_e(f_e^*) + (f_e - f_e^*)c_e(f_e)}.$$

Note that the definition of the Pigou bound accommodates both the cases $f_e^* \leq f_e$ and $f_e^* \geq f_e$. Rearranging,

$$f_e^* \cdot c_e(f_e^*) \geq \frac{1}{\alpha(\mathcal{C})} \cdot f_e \cdot c_e(f_e) + (f_e^* - f_e)c_e(f_e).$$

Summing this inequality over all $e \in E$ gives

$$C(f^*) \geq \frac{1}{\alpha(\mathcal{C})} \cdot C(f) + \underbrace{\sum_{e \in E} (f_e^* - f_e)c_e(f_e)}_{\geq 0 \text{ by (5)}} \geq \frac{C(f)}{\alpha(\mathcal{C})}.$$

Thus $C(f)/C(f^*) \leq \alpha(\mathcal{C})$, and the proof of Theorem 3.3 is complete.

References

[1] D. Braess. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung*, 12:258–268, 1968.

[2] J. R. Correa, A. S. Schulz, and N. E. Stier Moses. Selfish routing in capacitated networks. *Mathematics of Operations Research*, 29(4):961–976, 2004.

[3] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 1563 of *Lecture Notes in Computer Science*, pages 404–413, 1999.

[4] A. C. Pigou. *The Economics of Welfare*. Macmillan, 1920.

[5] T. Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67(2):341–364, 2003.

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

[7] Y. Sheffi. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, 1985.