Braess's Paradox, Fibonacci Numbers, and Exponential Inapproximability

Henry Lin[⋆], Tim Roughgarden[⋆⋆], Éva Tardos[⋆ ⋆ ⋆], and Asher Walkover[†]

Abstract. We give the first analyses in multicommodity networks of both the worst-case severity of Braess's Paradox and the price of anarchy of selfish routing with respect to the maximum latency. Our first main result is a construction of an infinite family of two-commodity networks, related to the Fibonacci numbers, in which both of these quantities grow exponentially with the size of the network. This construction has wide implications, and demonstrates that numerous existing analyses of selfish routing in single-commodity networks have no analogues in multicommodity networks, even in those with only two commodities. This dichotomy between single- and two-commodity networks is arguably quite unexpected, given the negligible dependence on the number of commodities of previous work on selfish routing.

Our second main result is an exponential upper bound on the worst-possible severity of Braess's Paradox and on the price of anarchy for the maximum latency, which essentially matches the lower bound when the number of commodities is constant.

Finally, we use our family of two-commodity networks to exhibit a natural network design problem with intrinsically exponential (in)approximability: while there is a polynomial-time algorithm with an exponential approximation ratio, subexponential approximation is unachievable in polynomial time (assuming $P \neq NP$).

1 Introduction

Selfish Routing and the Price of Anarchy A recent trend in theoretical computer science is to analyze the extent to which equilibria in a noncooperative game approximate a social optimum. The most popular measure is the *price of anarchy* [15] (also called the *coordination ratio* [13]), defined as the worst-case ratio between the objective function value of a Nash equilibrium of a game and that of an optimal solution. The price of anarchy is thus defined relative to a game and to an objective function.

In this paper, we will study one of the most popular models for price of anarchy analyses, the so-called *selfish routing* model. Selfish routing is a mathematical model of how noncooperative agents route traffic in a network with congestion. Formally, the game takes place in a directed multicommodity flow network, where each edge possesses a continuous, nondecreasing latency function that models how the performance of an edge degrades as it becomes increasingly congested. The traffic in the network is assumed to comprise a large number of independent network users, so that each individual has negligible impact on the experience of others. Under this assumption, equilibria—*flows at Nash equilibrium*—are naturally defined as the multicommodity flows in which all traffic travels only on minimum-latency paths.

As in most noncooperative games, flows at Nash equilibrium are inefficient, in the sense that they need not optimize natural objective functions. One such objective function, the average latency incurred by traffic, has been extensively studied. Beginning with Roughgarden and Tardos [21] and continuing with studies of ever-increasing generality [4, 6, 16, 18, 22], exact worst-case bounds on the price of anarchy with respect to the average latency have been established under a wide variety of different assumptions.

⋆ UC Berkeley, Computer Science Division, Soda Hall, Berkeley, CA 94720. Supported by a UC Berkeley Research Fellowship. Email: `henrylin@cs.berkeley.edu`.

⋆⋆ Department of Computer Science, Stanford University, 462 Gates Building, Stanford, CA 94305. Supported in part by ONR grant N00014-04-1-0725 and DARPA grant W911NF-04-9-0001. Email: `tim@cs.stanford.edu`.

⋆ ⋆ ⋆ Cornell University, Department of Computer Science, Upson Hall, Ithaca, NY 14853. Supported in part by NSF grant CCR-032553, NSF grant 0311333, and ONR grant N00014-98-1-0589. Email: `eva@cs.cornell.edu`.

† Google Inc., Mountain View, CA. Email: `walkover@gmail.com`.

The Maximum Latency Objective As in many combinatorial optimization problems, permitting an objective function to average the cost of different users can be problematic from a fairness perspective. Specifically, to attain or approximate a flow that minimizes the average latency, some users may need to be sacrificed to very costly paths, in order to reduce the congestion encountered by others. This unfairness inherent in the average latency measure motivates modifying the objective function to be more attuned to those users on the most costly paths. Arguably, the most obvious way to accomplish this is to aspire toward minimizing the *maximum* latency incurred by any user.

Compared to the average latency objective, considerably less is known about the price of anarchy relative to the maximum latency. The first paper on the topic is by Weitz [25], whose results we will review below. Most relevant for us is a paper by Roughgarden [19], where only the special case of single-commodity networks, networks in which all traffic shares the same source and destination, were considered. The main result of [19] states that, if latency functions are allowed to be arbitrary continuous, nondecreasing functions, then the (worst-case) price of anarchy with respect to the maximum latency objective in single-commodity networks with at most n vertices is precisely $n-1$.

Roughgarden [19] also made two conjectures about this price of anarchy in multicommodity networks. The *weak conjecture* of [19] asserts that in multicommodity networks, this price of anarchy can be bounded by a function of the number of vertices, edges, and commodities in the network. As a point of contrast, simple examples show that no such bound is possible for the price of anarchy relative to the average latency, unless additional structure is imposed on the network latency functions [21]. The *strong conjecture* of [19] states that the price of anarchy with respect to the maximum latency remains $n-1$ in multicommodity networks. This conjecture was motivated in part by the provable equivalence of single-commodity and multicommodity networks for the price of anarchy relative to the average latency [6, 18].

Braess's Paradox Braess's Paradox is the following counterintuitive fact: adding additional edges to a network with selfish routing can make all of the traffic worse off. First discovered in 1968 [3], Braess's Paradox has since motivated a vast number of follow-up papers; see [17] for a survey. Almost all existing work on the paradox confines attention to close variations on or analogues of Braess's original example in a four-node network. Only recently have larger, more severe versions of Braess's Paradox been discovered. Specifically, Roughgarden [17] defined an infinite family of networks, beginning with Braess's original example, that shows that adding edges to a single-commodity network with n vertices can increase the latency experienced by all of the traffic (and hence the maximum latency) by a factor of $\lfloor n/2 \rfloor$. It was also shown in [19] that no greater increase is possible in single-commodity networks, but the proof of this crucially used the combinatorial structure of flows at Nash equilibrium in such networks. Once again the story was left incomplete: are the networks of [17] the worst examples possible, or are more severe versions of Braess's Paradox lurking in the richer landscape of multicommodity networks?

Our Results In this paper, we establish nearly matching upper and lower bounds on both the price of anarchy with respect to the maximum latency and on the worst-possible severity of Braess's Paradox in multicommodity networks. Our results resolve both of the conjectures in [19]—one in the affirmative, one in the negative—and also give the first demonstration that Braess's Paradox is provably more severe in multicommodity networks than in single-commodity ones. Specifically, our two main results are the following.

- We give a parameterized construction, based on the Fibonacci numbers, that shows that adding one edge to a two-commodity network with n vertices can increase the latency of all traffic by a $2^{\Omega(n)}$ factor.
- We prove that the price of anarchy with respect to the maximum latency in networks with k commodities, n vertices, and m edges is $2^{O(\min\{kn, m \log n\})}$.

The construction used to prove the first result has wide implications. In particular, for all existing approximation-type analyses of selfish routing that were only known to hold for single-commodity networks [14, 17, 19], this construction rules out any reasonable extension to multicommodity networks, even those with only two commodities. For example, removing one edge from a single-commodity network can only decrease the maximum (or average) latency of a Nash flow by a factor of 2 [14], while our construction shows that a single edge removal can cause an exponential improvement in the average and the maximum latency (even with only two commodities). This dichotomy between single- and two-commodity networks is somewhat unexpected, given the negligible role that the number of commodities has played in previous work in this area [6, 18, 21].

The first result easily implies a lower bound of $2^{\Omega(n)}$ on the price of anarchy for the maximum latency in multicommodity networks, as an optimal flow has the option of ignoring edges that are causing Braess's Paradox. By the same reasoning, the second result implies that adding any number of edges to a network with k commodities, n vertices, and m edges can only increase the maximum latency by a $2^{O(\min\{kn, m \log n\})}$ factor. Our upper and lower bounds on both the price of anarchy and on the worst-possible severity of Braess's Paradox are thus essentially tight for networks with a constant number of commodities.

Finally, we consider the following network design problem, motivated by the goal of detecting and avoiding Braess's Paradox: given a network, find the subnetwork with the smallest maximum latency. Using our family of two-commodity networks and ideas from the gap reductions of [17] that apply to the single-commodity version of the problem, we prove that there is no polynomial-time algorithm for this network design problem with subexponential approximation ratio (assuming $P \neq NP$). Since our upper bound on the price of anarchy trivially implies that an exponential performance guarantee is achievable, this network design problem is a rare example of a natural optimization problem with intrinsically exponential approximability.

Further Related Work There have been numerous price of anarchy analyses in the past few years. Study of the original load-balancing model of Koutsoupias and Papadimitriou [13] continues unabated; see [7, 10] for surveys. A survey of the selfish routing model studied here, including results on the price of anarchy, can be found in [20]. Other noncooperative games have also been studied recently from a price of anarchy perspective, including facility location games [8, 24], network design games [1, 9], and resource allocation games [12].

As noted above, Weitz [25] was the first to study the price of anarchy of selfish routing under the maximum latency objective. Weitz [25] noted that, for single-commodity networks and classes of restricted latency functions, the price of anarchy for the maximum latency is no more than that for the average latency objective. For example, a theorem of Roughgarden and Tardos [21] bounding the price of anarchy with respect to the average latency objective then implies that the price of anarchy for the maximum latency in single-commodity networks with linear latency functions is at most $4/3$, and a matching lower bound is furnished by the original form of Braess's Paradox [3, 25]. However, upper bounds on the price of anarchy with respect to the maximum latency objective do not imply upper bounds on the price of anarchy with respect to the average latency objective: for example, the price of anarchy for the maximum latency objective is at most $n-1$ in single-commodity networks with arbitrary latency functions [19], while the price of anarchy for the average latency can be arbitrarily large even in two-node, two-link networks [21].

Weitz [25] also gave a family of networks that shows that this price of anarchy is $\Omega(n)$ for multicommodity networks with n vertices and linear latency functions. Concurrently with Roughgarden [19], Correa, Schulz, and Stier Moses [5] studied the maximum latency objective from several different perspectives. The results of [5] mostly concern the computational complexity of computing an optimal solution and the extent to which multiple objective functions can be simultaneously optimized, and are disjoint from those in [19] and in the present work.

2 Preliminaries

The Model We now describe our model of selfish routing, following Roughgarden and Tardos [21]. We will study a multicommodity flow network, described by a directed graph $G = (V, E)$ and k source-destination vertex pairs $(s_1, t_1), \ldots, (s_k, t_k)$. We denote by r_i the amount of traffic that wishes to travel from the source s_i to the destination t_i—the *traffic rate*. The graph G can contain parallel edges, but we can exclude self-loops. We will denote the s_i-t_i paths of G by $\mathcal{P}_i$. We assume that $\mathcal{P}_i$ is non-empty for all i, and define $\mathcal{P} = \cup_{i=1}^{k} \mathcal{P}_i$.

A *flow* is a nonnegative vector indexed by $\mathcal{P}$. By f_e we mean the amount $\sum_{P \in \mathcal{P} \,:\, e \in P} f_P$ of flow that traverses edge e. With respect to a network G and a vector r of traffic rates, a flow is *feasible* if $\sum_{P \in \mathcal{P}_i} f_P = r_i$ for all commodities i.

We assume that the network G suffers from congestion effects, and to model this we give edge e a nonnegative, continuous, nondecreasing *latency function* ℓ_e that describes the time needed to traverse the edge as a function of the edge congestion f_e. Given a flow f, the latency ℓ_P of a path P is the sum of the latencies of the edges in the path: $\ell_P(f) = \sum_{e \in P} \ell_e(f_e)$. We will call a triple of the form (G, r, ℓ) an *instance*.

Our objective function is the maximum latency incurred by a flow, defined formally by $M(f) = \max_{P \in \mathcal{P} \,:\, f_P > 0} \ell_P(f)$. With respect to an instance (G, r, ℓ), a flow that minimizes $M(\cdot)$ over all feasible flows will be called *optimal*. Since the feasible flows of an instance form a compact subset of Euclidean space and $M(\cdot)$ is a continuous function, every instance admits an optimal flow.

Flows at Nash Equilibrium We next define the flows that we expect to arise from selfish routing. Assuming that all network users have negligible size and want to minimize the latency experienced, we expect all users to travel on paths with minimum-possible latency. We formalize this in the next definition.

Definition 1. *A flow f feasible for (G, r, ℓ) is* at Nash equilibrium*, or is a* Nash flow*, if for every $i \in \{1, 2, \ldots, k\}$ and two paths $P_1, P_2 \in \mathcal{P}_i$ with $f_{P_1} > 0$,*

$$\ell_{P_1}(f) \leq \ell_{P_2}(f).$$

Happily, Nash flows always exist, and all Nash flows of an instance have equal maximum latency.

Proposition 1 *Let (G, r, ℓ) be an instance.*

(a) There is at least one Nash flow for (G, r, ℓ).
(b) If $f, \tilde{f}$ are Nash flows for (G, r, ℓ), then $M(f) = M(\tilde{f})$.

Proposition 1 is classical; for example, it follows from arguments of Beckmann, McGuire, and Winsten [2].

Definition 1 implies that in a Nash flow, all of the traffic of a given commodity experiences a common latency. We will sometimes use the notation $L_i(G, r, \ell)$ to denote the common latency of the ith commodity's traffic in a Nash flow for (G, r, ℓ); analogously to Proposition 1(b), this is well defined (i.e., independent of the particular Nash flow).

We will also benefit from the following alternative definition of a Nash flow, which was first noted by Smith [23]. It is an easy consequence of Definition 1.

Proposition 2 *A flow f feasible for (G, r, ℓ) is at Nash equilibrium if and only if*

$$\sum_{e \in E} \ell_e(f_e) f_e \leq \sum_{e \in E} \ell_e(f_e) \tilde{f}_e \tag{1}$$

for every flow $\tilde{f}$ that is feasible for (G, r, ℓ).

The Price of Anarchy We now formalize what we mean by the price of anarchy. As noted in the introduction, it is the ratio of the objective function values of a flow at Nash equilibrium and an optimal flow. If (G, r, ℓ) is an instance, then the *price of anarchy of* (G, r, ℓ), denoted $\rho(G, r, \ell)$, is the ratio $M(f)/M(f^*)$, where f is a Nash flow and f^* is an optimal flow. Proposition 1 ensures that the price of anarchy of an instance is well defined provided $M(f^*) > 0$. If $M(f^*) = 0$, then f^* is also a flow at Nash equilibrium and we define the price of anarchy of the instance to be 1.

Finally, the price of anarchy $\rho(\mathcal{I})$ of a collection $\mathcal{I}$ of instances is defined in the obvious way:

$$\rho(\mathcal{I}) = \sup_{(G,r,\ell) \in \mathcal{I}} \rho(G, r, \ell).$$

3 Braess's Paradox in Multicommodity Networks

In this section, we prove that Braess's Paradox can be much more severe in multicommodity networks than in single-commodity networks. In fact, there will be a "phase transition" of sorts: the worst-case severity of Braess's Paradox is polynomial in single-commodity instances, but exponential in two-commodity instances. The family of instances that we construct in this section will also serve as a starting point for our inapproximability results in Section 5.

We will begin this section by formally stating the properties of our construction in Theorem 3 below. Prior to detailing this construction and proving Theorem 3, we will discuss its many consequences for multicommodity networks.

Our family of two-commodity instances is closely related to the *Fibonacci numbers*. Recall that for a nonnegative integer p, the pth Fibonacci number F_p is defined as follows: $F_0 = 0$, $F_1 = 1$, and $F_p = F_{p-2} + F_{p-1}$ for $p \geq 2$. It is well known that $F_p \approx c \cdot \phi^p$ as $p \to \infty$, where $c \approx 0.4472$ and $\phi \approx 1.618$ is the golden ratio.

We can now state the main result of this section.

Theorem 3. *There is an infinite family $\{(G^p, r^p, \ell^p)\}_{p=1}^{\infty}$ of instances with the following properties:*

(a) (G^p, r^p, ℓ^p) *has two commodities and* $O(p)$ *vertices and edges as* $p \to \infty$*;*
(b) for p *odd,* $L_1(G^p, r^p, \ell^p) = F_{p-1} + 1$ *and* $L_2(G^p, r^p, \ell^p) = F_p$*;*
(c) for p *even,* $L_1(G^p, r^p, \ell^p) = F_p + 1$ *and* $L_2(G^p, r^p, \ell^p) = F_{p-1}$*;*
(d) for all p*, there is a subgraph* H^p *of* G^p *with one less edge than* G^p *that satisfies* $L_1(H^p, r^p, \ell^p) = 1$ *and* $L_2(H^p, r^p, \ell^p) = 0$*.*

Theorem 3 has a number of implications. We begin by noting two immediate corollaries of the theorem.

Corollary 1. *Adding a single edge to an* n*-vertex two-commodity instance can increase the latency of all traffic by a* $2^{\Omega(n)}$ *factor as* $n \to \infty$*.*

Corollary 2. *If* $\mathcal{I}_n$ *is the set of instances with at most* n *vertices, then* $\rho(\mathcal{I}_n) = 2^{\Omega(n)}$ *as* $n \to \infty$*.*

Furthermore, Corollary 1 trivially implies that for every $k \geq 2$, adding a single edge to an n-vertex k-commodity instance can increase the latency of all traffic by a $2^{\Omega(n)}$ factor as $n \to \infty$.

Theorem 3 and Corollaries 1 and 2 show that a number of previously established properties of single-commodity instances do not carry over to multicommodity networks. In particular, the following statements are known to hold in single-commodity instances.

(1) Adding one edge to a single-commodity instance can only increase the maximum or average latency of a Nash flow by a factor of 2 [14].
(2) Adding any number of edges to an n-vertex single-commodity instance can only increase the maximum or average latency of a Nash flow by a factor of $\lfloor n/2 \rfloor$ [17].
(3) The price of anarchy with respect to maximum latency in an n-vertex single-commodity instance is at most $n - 1$ [19].

Theorem 3 and Corollaries 1 and 2 demonstrate that all of these statements utterly fail to extend to multicommodity networks, even to those with only two commodities. This dichotomy stands in contrast to other work on selfish routing, such as bounds on the price of anarchy with respect to the average latency objective function, where there is provably no separation between single-commodity and multicommodity instances [18, 6].

We now give the construction of the family of instances claimed in Theorem 3. We begin by defining the graph G^p for $p \geq 1$, see Figure 1. We will describe the construction only for p odd; the construction for even p is similar. We begin with two paths, which we will call P_1 and P_2. The $(p+3)$-vertex path P_2, drawn vertically in Figure 1, is $s_2 \to w_0 \to w_1 \to \cdots \to w_p \to t_2$. The $(p+4)$-vertex path P_1, drawn horizontally in Figure 1, is $s_1 \to a \to w_1 \to v_1 \to \cdots \to v_p \to t_1$. We also add the following edges between the two paths:

- (a, w_i) for all positive even i;
- (v_i, w_i) for all positive even i;
- (s_2, v_i) for all odd i at most $p - 2$;
- (w_i, v_i) for all odd i.

Finally, we complete G^p by adding what we will call an *extra edge*, defined as the edge (s_1, w_0).

For all p, the traffic rate vector r^p will be $r_1^p = r_2^p = 1$. To complete the construction, we therefore need only describe the edge latency functions. All edges will either possess a constant latency function, or a latency function that approximates a step function. We next introduce notation for the latter function type. For a positive integer i and a positive real number δ, f_δ^i will denote a continuous, nondecreasing function satisfying $f_\delta^i(x) = 0$ for $x \leq 1$ and $f_\delta^i(x) = F_i$ for $x \geq 1 + \delta$. (The function f_δ^i can be defined arbitrarily on $(1, 1 + \delta)$, provided it is continuous and nondecreasing.)

For $i \in \{0, 1, \ldots, p-1\}$, we define the edge e_i to be (w_i, w_{i+1}) if i is even and (v_i, v_{i+1}) if i is odd. (See Figure 1.) We now define the latency functions ℓ^p for G^p as follows, where δ is sufficiently small (to be chosen later): for each $i > 0$, edge e_i receives the latency function $\ell^p(x) = f_\delta^i(x)$, edge e_0 receives the latency function $\ell^p(x) = f_\delta^1(x)$, edge (s_1, a) receives the latency function $\ell^p(x) = 1$, and all other edges receive the latency function $\ell^p(x) = 0$.

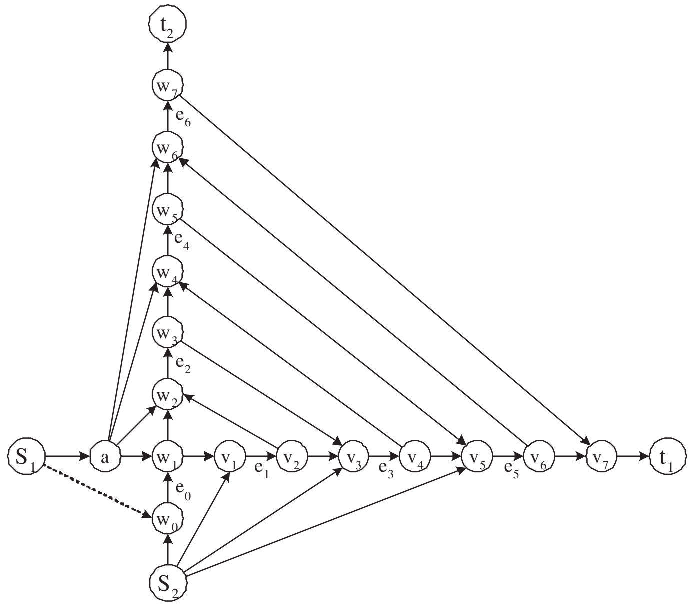

Fig. 1. Construction of the instance (G^p, r^p, ℓ^p) when $p = 7$. Dotted edge (s_1, w_0) is the "extra edge". Edges with non-constant latency functions are labelled.

With the construction in hand, we now turn toward proving Theorem 3 for odd p (the arguments for even p are similar). Part (a) is obvious. Part (d) is easy to see: if H^p is obtained from G^p by removing the extra edge (s_1, w_0) and f is the flow that routes one unit of traffic on both P_1 and P_2, then f is at Nash equilibrium for (G^p, r^p, ℓ^p), showing that $L_1(H^p, r^p, \ell^p) = 1$ and $L_2(H^p, r^p, \ell^p) = 0$. (See Figure 2.)

To finish the proof of Theorem 3 (for p odd), we need only prove part (b). We will accomplish this via a sequence of lemmas, the first of which requires some further definitions. First, we will say that a flow f, feasible for (G^p, r^p, ℓ^p), *floods* the instance if $f_{e_i} \geq 1 + \delta$ for all $i \in \{0, 1, \ldots, p-1\}$. Thus if f floods (G^p, r^p, ℓ^p), all edge latencies are at their maximum, as in Figure 3. Second, we introduce notation for some of the paths of G^p. For i even, Q_i will denote the unique s_1-t_1 path which traverses edge e_i before any odd labelled edges, and includes no other edge of P_2. For i odd, Q_i will denote the unique s_2-t_2 path which traverses edge e_i before any even labelled edges, and includes no other edge of P_1. We will call the paths $Q_0, \ldots, Q_{p-1}$, together with the "axis-aligned" paths P_1 and P_2, the *short paths*. The next lemma justifies this terminology, at least for flows that flood the instance (G^p, r^p, ℓ^p).

Lemma 1. *If f floods (G^p, r^p, ℓ^p) with p odd, then:*

(a) $\ell_P(f) \geq F_{p-1} + 1$ for every s_1-t_1 path P, and equality holds for short paths;
(b) $\ell_P(f) \geq F_p$ for every s_2-t_2 path P, and equality holds for short paths.

We will only prove part (b) of Lemma 1, as the proof of part (a) is similar. In the proof, we will use the following lemma about Fibonacci numbers, which is easy to verify.

Lemma 2. *Let j and p be odd positive integers with $j < p$, and I the even numbers between j and p. Then, $F_j + \sum_{i \in I} F_i = F_p$.*

We now prove Lemma 1.

Proof of Lemma 1: Let P be an s_2-t_2 path. Let j be the largest odd number such that $e_j \in P$, or 0 if there is no such number. We only need to prove the case where $j > 0$, since the $j = 0$ and $j = 1$ cases are the same. If $j > 0$, then P contains e_j and also e_i for all even i between j and p. Since f floods (G^p, r^p, ℓ^p), Lemma 2 implies that $\ell_P(f) \geq F_p$. Moreover, this inequality holds with equality for short paths. ■

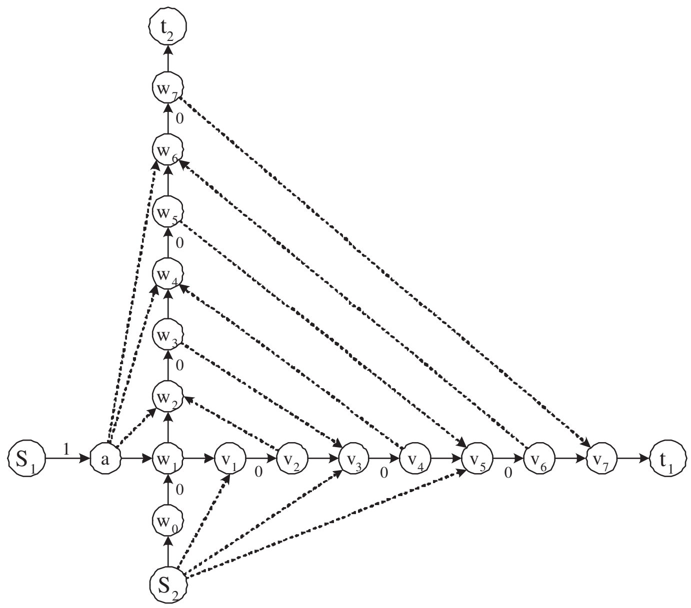

Fig. 2. Nash flow in (H^p, r^p, ℓ^p), with $p = 7$, where H^p is G^p with the extra edge (s_1, w_0) removed. Solid edges carry flow, dotted edges do not. Edge latencies are with respect to the Nash flow. Unlabelled edges have zero latency.

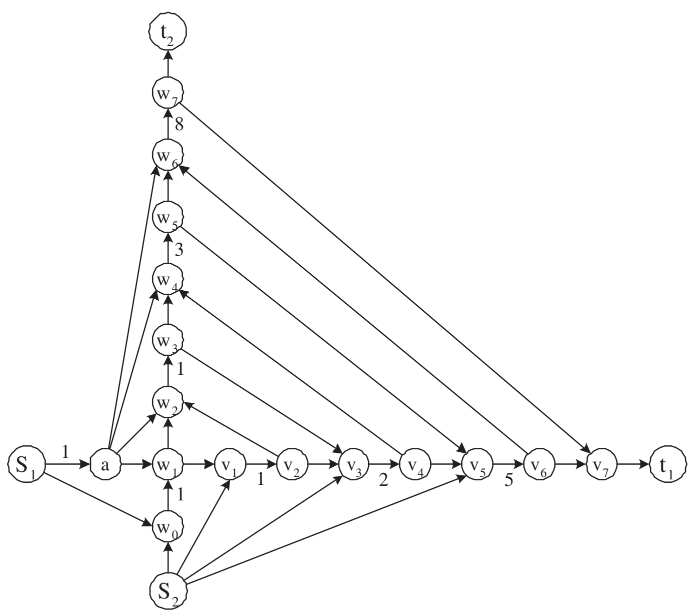

Fig. 3. Nash flow in (G^p, r^p, ℓ^p), with $p = 7$. Solid edges carry flow, dotted edges do not. Edge latencies are with respect to the Nash flow. Unlabelled edges have zero latency.

Our final lemma states that routing flow on short paths suffices to flood the instance (G^p, r^p, ℓ^p). For the statement of the lemma, recall that the parameter δ controls how rapidly the non-constant latency functions of (G^p, r^p, ℓ^p) increase as the amount of flow on the edge exceeds one.

Lemma 3. *For all p odd and δ sufficiently small, there is flow f, with $f_P > 0$ only for short paths P, that floods (G^p, r^p, ℓ^p).*

Proof. Define the flow f as follows. First, for $i = 0, 1, \ldots, p-1$, route $2^{-(i+1)}$ units of flow (of the appropriate commodity) on the short path Q_i. This routes strictly less than one unit of flow of each commodity. The remaining flow is then routed on the short paths P_1 and P_2.

To complete the proof, we need to show that f floods (G^p, r^p, ℓ^p)—that $f_{e_i} \geq 1 + \delta$ for all $p \in \{0, 1, \ldots, p-1\}$ provided δ is sufficiently small. We will prove this inequality only for i odd; the argument for even i is similar.

The second commodity uses edge e_i only in the short path Q_i, on which it routes $2^{-(i+1)}$ units of flow. The first commodity uses edge e_i in all of its flow paths except for the short paths Q_j for j even and greater than i. The total amount of flow on e_i is thus at least

$$2^{-(i+1)} + 1 - \sum_{j \geq 0} 2^{-(i+2+2j)} = 1 + 2^{-(i+1)} - \frac{4}{3} \cdot 2^{-(i+2)} > 1 + 2^{-(i+3)}.$$

Thus, provided $\delta \leq 2^{-(p+3)}$, f floods (G^p, r^p, ℓ^p), and the proof is complete.

Theorem 3(b) now follows immediately from Definition 1, Lemma 1, and Lemma 3.

4 Upper Bounds on the Price of Anarchy

We now turn toward proving upper bounds on the price of anarchy and, as a consequence, on the worst-possible severity of Braess's Paradox. We will aim for an upper bound that matches the lower bound of Theorem 3, and will largely succeed in this goal.

We begin by proving a very weak bound on the price of anarchy, a bound that depends on parameters other than the size of the network. While not interesting in its own right, this bound will play a crucial role in later proofs in this section.

Lemma 4. *Let f be a Nash flow and f^* a feasible flow for an instance (G, r, ℓ), where G has m edges. For every edge e of G with $f_e > f^*_e$,*

$$\ell_e(f_e) \leq \frac{m \sum_i r_i}{f_e - f^*_e} \cdot \max_e \ell_e(f^*_e). \tag{2}$$

Proof of Lemma 4: Let $F \subseteq E$ denote the edges e of G for which $f_e > f^*_e$. Using inequality (1) in Proposition 2 and the fact that $\ell_e(f^*_e) \leq M(f^*)$ whenever $f^*_e > 0$, we can derive the following crude bound:

$$\begin{aligned} \sum_{e \in F} \ell_e(f_e)(f_e - f^*_e) &\leq \sum_{e \in E \setminus F} \ell_e(f_e)(f^*_e - f_e) \\ &\leq \left(\max_e \ell_e(f^*_e)\right) \sum_{e \in E \setminus F} (f^*_e - f_e) \\ &\leq \max_e \ell_e(f^*_e) \cdot m \cdot \sum_i r_i. \end{aligned}$$

The lemma now follows easily. ■

We next use Lemma 4 as a bootstrap for deriving upper bounds on the price of anarchy that depend only on the size of the network. We will accomplish this as follows. For an arbitrary instance, we will set up a linear program, with edge latencies as variables, that maximizes the price of anarchy among instances that are "basically equivalent" to the given instance. We will define our notion of equivalence so that Lemma 4 ensures that the linear program has a bounded maximum, and will then analyze the vertices of the feasible region of the linear program to derive the following bound.

Theorem 4. *If (G, r, ℓ) is an instance with n vertices and m edges, then*

$$\rho(G, r, \ell) = 2^{O(m \log n)}.$$

Before implementing the proof approach outlined above, we state a proposition that bounds the maximum size of the optimal value of a linear program with a constraint matrix with entries in $\{-1, 0, 1\}$.

Proposition 5 *Let A be an $m \times n$ matrix with entries in $\{0, \pm 1\}$ and at most α non-zero entries in each row. Let b be a real-valued m-vector, and let the linear program $\max x_i$ subject to $Ax \le b$ have a finite maximum. Then, this maximum is at most $n\alpha^n \|b\|_\infty$, where $\|b\|_\infty$ denotes $\max_j |b_j|$.*

Proposition 5 can be proved with Cramer's rule and a simple bound on the determinant. We omit further details.

Proof of Theorem 4: Let (G, r, ℓ) be an instance with n vertices and m edges. Let f and f^* be Nash and optimal flows for (G, r, ℓ), respectively. We aim to show that $\rho(G, r, \ell) = 2^{O(m \log n)}$.

We begin by performing some preprocessing on the instance (G, r, ℓ). First, if $f_e = f_e^* = 0$ for some edge e, then that edge can be removed from the instance without affecting its ρ-value. We can therefore assume that $f_e^* > 0$ or $f_e > 0$ for every edge e. Second, we can assume that $\ell_e(0) = 0$ whenever $f_e^* = 0$. To see why, note that replacing the latency function $\ell_e(x)$ of such an edge by the function equal to (e.g.) $\min\{x/f_e, 1\} \cdot \ell_e(x)$ leaves the Nash flow unaffected while only decreasing the maximum latency of f^* and hence increasing the ρ-value of the instance. Combining these two assumptions, we can assume without loss of generality $\ell_e(f_e^*) \le M(f^*)$ for every edge e of G.

We now set up a linear program that attempts to further transform the latency functions to make the ρ-value of the given instance as large as possible. In the linear program, the flow amounts $\{f_e\}$ and $\{f_e^*\}$, as well as the latencies $\{\ell_e(f_e^*)\}$ with respect to f^*, will be held fixed. There will be a nonnegative variable $\hat{\ell}_e(f_e)$ representing the latency of edge e with respect to the flow f. So that the new latency functions are nondecreasing, we impose the following linear constraints, which we call *monotonicity constraints*:

- For all edges e with $f_e = f_e^*$, $\hat{\ell}_e(f_e) = \ell_e(f_e^*)$.
- For all edges e with $f_e < f_e^*$, $\hat{\ell}_e(f_e) \le \ell_e(f_e^*)$.
- For all edges e with $f_e > f_e^*$, $\hat{\ell}_e(f_e) \ge \ell_e(f_e^*)$.

Additionally, we will insist that the (fixed) flow f be at Nash equilibrium with respect to the (variable) latencies $\{\hat{\ell}_e(f_e)\}$. There are several ways that this requirement can be encoded with linear constraints. For this proof, we will be content with the following naive approach: for every commodity i, and every pair of paths $P, \tilde{P} \in \mathcal{P}_i$ for which $f_e^{(i)} > 0$ for all $e \in P$, we insist that $\sum_{e \in P} \hat{\ell}_e(f_e) \le \sum_{e \in \tilde{P}} \hat{\ell}_e(f_e)$ in our linear program. Since this linear program has a small number of variables, we will not be hampered by its potentially massive number of constraints.

By construction, our constraints ensure the following: for every feasible solution $\{\hat{\ell}(f_e)\}$, there is an instance $(G, r, \hat{\ell})$ with continuous, nondecreasing latency functions $\hat{\ell}$, so that these latency functions interpolate their two prescribed values and f is a Nash flow for $(G, r, \hat{\ell})$. Consider the objective function $\max \hat{\ell}_e(f_e)$ for an edge e. Our key claim is that the resulting linear program is not unbounded. For edges e with $f_e \le f_e^*$, the claim is obvious from the constraints. For edges e with $f_e > f_e^*$, the claim follows from Lemma 4 and the fact that all parameters on the right-hand side of the bound (2) are fixed in the linear program.

Since the maximum of the above linear program is bounded, we can apply Proposition 5. In our linear program, there are a total of m variables, of which each constraint contains at most $2n$. The right-hand side of each constraint is either a 0 or a term of the form $\ell_e(f_e^*)$. By our preprocessing step, $\ell_e(f_e^*) \le M(f^*)$ for all edges e. Hence, Proposition 5 implies that the maximum of the linear program is at most $mn^{O(m)} \cdot M(f^*)$. Hence, returning to the original instance (G, r, ℓ), we must have $\ell_e(f_e) \le mn^{O(m)} \cdot M(f^*)$ for all edges e. Since a flow path of f can contain only n edges, we can conclude that $\rho(G, r, \ell) \le nmn^{O(m)} = 2^{O(m \log n)}$. ■

When the number of commodities is small (e.g., a constant), we can improve the bound to $2^{O(kn)}$ using a different way to encode the constraint that f must be at Nash equilibrium with respect to $\{\hat{\ell}_e(f_e)\}$.

Theorem 6. *If (G, r, ℓ) is an instance with n vertices and k commodities, then*

$$\rho(G, r, \ell) = 2^{O(kn)}.$$

Proof. To prove our bound, we start with the linear program described in Theorem 4. We leave the objective and monotonicity constraints the same, but replace the constraints that ensure f is a Nash equilibrium for $\{\hat{\ell}_e(f_e)\}$. To ensure the latencies $\{\hat{\ell}_e(f_e)\}$ define a Nash equilibrium for f, we introduce an auxiliary variable $\hat{d}^i(v)$ for each commodity i and for every vertex $v \in V$ reachable from that commodity's source s_i, which will represent the length of the shortest path from s_i to v, with respect to the latencies $\{\hat{\ell}_e(f_e)\}$. Now, we define the following constraints:

- $\hat{d}_i(s_i) = 0$, for all commodities i.
- $\hat{d}_i(u) + \hat{\ell}_e(f_e) = \hat{d}_i(v)$, for all edges $e = (u, v)$ and commodities i with $f_e^{(i)} > 0$.
- $\hat{d}_i(u) + \hat{\ell}_e(f_e) \leq \hat{d}_i(v)$, for all edges $e = (u, v)$ and commodities i.

To complete the proof, we need to show:

(a) A set of latencies $\{\hat{\ell}_e(f_e)\}$ is feasible for our linear program if and only if it defines a Nash equilibrium for f.
(b) The latency variables can be removed, yielding a linear program with kn variables, that can be bounded by $2^{O(kn)}$.

To prove the forward direction of (a), it is easy to see that with our constraints, for every commodity i, each path $P \in \mathcal{P}_i$, with $f_e^{(i)} > 0$ for all $e \in P$, must have length precisely equal to $\hat{d}_i(t_i)$. Furthermore, no path in $P \in \mathcal{P}_i$ may have length strictly less than $\hat{d}_i(t_i)$. Therefore, for every commodity i and every $P, \tilde{P} \in \mathcal{P}_i$, with $f_e^{(i)} > 0$ for all $e \in P$, $\sum_{e \in P} \hat{\ell}_e(f_e) \leq \sum_{e \in \tilde{P}} \hat{\ell}_e(f_e)$, and our latencies define a Nash equilibrium for f. To prove the other direction of (a), note that for any set of latencies $\hat{\ell}_e(f_e)$ defining a Nash equilibrium for f, we can define $\hat{d}_i(v)$ to be the length of the shortest path from s_i to v, and we have a feasible solution for our linear program.

With (a) proven, we know that the maximum value of our linear program is precisely the maximum edge length that occurs in any Nash equilibrium for flow f. As before, note that Lemma 4 implies our linear program is bounded, and we can apply Proposition 5 to bound the maximum value of the linear program.

Before applying Proposition 5 however, we first eliminate the latency variables, which allows us to prove a better bound. Note that for any edge $e = (u, v)$ with $f_e^i > 0$ for some commodity i, $\hat{d}_i(u) + \hat{\ell}_e(f_e) = \hat{d}_i(v)$, so we can replace any occurrence of $\hat{\ell}_e(f_e)$ in our linear program with $\hat{d}_i(v) - \hat{d}_i(u)$. Furthermore, for any other edges, $f_e = 0$, and it must be the case that $\hat{\ell}_e(f_e) \leq M(f^*)$. For these edges, an optimal solution must assign some value $x_e \leq M(f^*)$ to the latencies $\hat{\ell}_e(f_e)$, and thus for these edges, we can substitute the $\hat{\ell}_e(f_e)$ variable with the constant x_e value used in the optimal solution. With these substitutions, we have not changed the optimal value of our linear program, and we are only left with $O(kn)$ variables. Moreover, there are still a constant number of variables per constraint, and each entry of b is still bounded by $M(f^*)$. Therefore, applying Proposition 5 bounds the price of anarchy by $2^{O(kn)}$.

Corollary 2 shows that Theorem 6 is essentially tight for a constant number of commodities.

5 Exponential Inapproximability for Network Design

In this section, we will show that a network design problem that is naturally motivated by Braess's Paradox has intrinsically exponential approximability. The problem, which we call MULTICOMMODITY NETWORK DESIGN (MCND), is as follows.

Given a (multicommodity) instance (G, r, ℓ), find a subgraph H of G that minimizes $M(H, r, \ell)$.

By $M(H, r, \ell)$, we mean the maximum latency of a Nash flow for (H, r, ℓ) (well defined by Proposition 1). *MCND* is tantamount to detecting and avoiding Braess's Paradox. For single-commodity instances, this problem was studied in [17].

The *trivial algorithm* is defined as the algorithm that always returns the entire graph G—the algorithm that always punts on trying to detect Braess's Paradox. The following was proved in [17]: the trivial algorithm is an $\lfloor n/2 \rfloor$-approximation algorithm for the special case of single-commodity instances; and for every $\epsilon > 0$, no $(\lfloor n/2 \rfloor - \epsilon)$-approximation algorithm exists (assuming $P \neq NP$). Here, we will succeed in proving analogues of these results for multicommodity networks, where the best-possible approximation ratio is inherently exponential.

First, we note that since Theorems 4 and 6 imply limits on the largest possible increase in the maximum latency due to Braess's Paradox, they also translate to an upper bound on the trivial algorithm.

Proposition 7 *The trivial algorithm is a* $2^{O(\min\{kn, m\log n\})}$*-approximation algorithm for* MCND.

Much more interesting is the next result, which states that there is *no* polynomial-time algorithm with subexponential approximation ratio (assuming $P \neq NP$).

Theorem 8. *Assuming* $P \neq NP$*, there is no* $2^{o(n)}$*-approximation algorithm for* MCND.

The proof of Theorem 8 combines ideas from the gap reductions of [17] for the single-commodity version of MCND with the family of two-commodity instances described in Section 3. Because of space constraints, we will content ourselves here with a high-level overview of the proof.

Recall that in an instance of the NP-complete problem PARTITION, we are given q positive integers $\{a_1, a_2, \ldots, a_q\}$ and seek a subset $S \subseteq \{1, 2, \ldots, q\}$ such that $\sum_{j \in S} a_j = \frac{1}{2}\sum_{j=1}^{q} a_j$ [11, SP12]. The idea of the reduction is to start with an instance (G^p, r^p, ℓ^p) of the form described in Section 3, and to replace the extra edge (s_1, w_0) with a collection of parallel edges representing an instance $\mathcal{I} = \{a_1, \ldots, a_p\}$ of PARTITION. We will give these edges latency functions that simulate "capacities", with an edge representing an integer a_j of $\mathcal{I}$ receiving capacity proportional to a_j. The proof then break down into three parts. First, if too many of these parallel edges are removed from the network, there will be insufficient remaining capacity to send flow cheaply. To implement this, we must also augment the latency functions of the edges $e_0, \ldots, e_{p-1}$ of Section 3 to have effective capacities. Second, if too few of the parallel edges are removed, the excess of capacity results in a bad flow at Nash equilibrium similar to that of Figure 3. Finally, these two cases can be avoided if and only if $\mathcal{I}$ is a "yes" instance of PARTITION, in which case removing the appropriate collection of parallel edges results in a network that admits a good Nash equilibrium similar to that of Figure 2.

References

1. E. Anshelevich, A. Dasgupta, É. Tardos, and T. Wexler. Near-optimal network design with selfish agents. In *Proceedings of the 35th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 511–520, 2003.
2. M. Beckmann, C. B. McGuire, and C. B. Winsten. *Studies in the Economics of Transportation*. Yale University Press, 1956.
3. D. Braess. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung*, 12:258–268, 1968.
4. C. K. Chau and K. M. Sim. The price of anarchy for non-atomic congestion games with symmetric cost maps and elastic demands. *Operations Research Letters*, 31(5):327–335, 2003.
5. J. R. Correa, A. S. Schulz, and N. E. Stier Moses. Computational complexity, fairness, and the price of anarchy of the maximum latency problem. In *Proceedings of the 10th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, volume 3064 of *Lecture Notes in Computer Science*, pages 59–73, 2004.
6. 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.
7. A. Czumaj. Selfish routing on the Internet. In J. Leung, editor, *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, chapter 42. CRC Press, 2004.
8. N. Devanur, N. Garg, R. Khandekar, V. Pandit, and A. Saberi. Price of anarchy, locality gap, and a network service provider game. Unpublished manuscript, 2003.
9. A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. J. Shenker. On a network creation game. In *Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 347–351, 2003.
10. R. Feldmann, M. Gairing, T. Lücking, B. Monien, and M. Rode. Selfish routing in non-cooperative networks: A survey. In *Proceedings of the Conference on Mathematical Foundations of Computer Science (MFCS)*, volume 2747 of *Lecture Notes in Computer Science*, pages 21–45, 2003.
11. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.
12. R. Johari and J. N. Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research*, 29(3):407–435, 2004.
13. E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 404–413, 1999.
14. H. Lin, T. Roughgarden, and É. Tardos. A stronger bound on braess's paradox. In *Proceedings of the 15th Annual Symposium on Discrete Algorithms (SODA)*, pages 333–334, 2004.
15. C. H. Papadimitriou. Algorithms, games, and the Internet. In *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing (STOC)*, pages 749–753, 2001.
16. G. Perakis. The price of anarchy when costs are non-separable and asymmetric. In *Proceedings of the 10th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, volume 3064 of *Lecture Notes in Computer Science*, pages 46–58, 2004.

17. T. Roughgarden. Designing networks for selfish users is hard. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, pages 472–481, 2001.
18. T. Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67(2):341–364, 2003.
19. T. Roughgarden. The maximum latency of selfish routing. In *Proceedings of the 15th Annual Symposium on Discrete Algorithms (SODA)*, pages 973–974, 2004.
20. T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, 2005.
21. T. Roughgarden and É. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.
22. T. Roughgarden and É. Tardos. Bounding the inefficiency of equilibria in nonatomic congestion games. *Games and Economic Behavior*, 47(2):389–403, 2004.
23. M. J. Smith. The existence, uniqueness and stability of traffic equilibria. *Transportation Research*, 13B:295–304, 1979.
24. A. Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 416–425, 2002.
25. D. Weitz. The price of anarchy. Unpublished manuscript, 2001.