SIAM J. DISCRETE MATH.
Vol. 25, No. 4, pp. 1667–1686

STRONGER BOUNDS ON BRAESS'S PARADOX AND THE MAXIMUM LATENCY OF SELFISH ROUTING*

HENRY LIN†, TIM ROUGHGARDEN‡, ÉVA TARDOS§, AND ASHER WALKOVER¶

Abstract. We give several new upper and lower bounds on the worst-case severity of Braess's paradox and the price of anarchy of selfish routing with respect to the maximum latency objective. In single-commodity networks with arbitrary continuous and nondecreasing latency functions, we prove that this worst-case price of anarchy is exactly $n - 1$, where n is the number of network vertices. For Braess's paradox in such networks, we prove that removing at most c edges from a network decreases the common latency incurred by traffic at equilibrium by at most a factor of $c + 1$. In particular, the worst-case severity of Braess's paradox with a single edge removal is maximized in Braess's original four-vertex network. In multicommodity networks, we exhibit an infinite family of two-commodity networks, related to the Fibonacci numbers, in which both the worst-case severity of Braess's paradox and the price of anarchy for the maximum latency objective grow exponentially with the network size. This construction demonstrates that numerous known selfish routing results for single-commodity networks have no analogues in networks with two or more commodities. We also prove an upper bound on both of these quantities that is exponential in the network size and independent of the network latency functions, showing that our construction is close to optimal. Finally, we use our family of two-commodity networks to exhibit a natural network design problem with intrinsically exponential (in)approximability.

Key words. Braess's paradox, selfish routing, price of anarchy, traffic networks, approximation algorithms

AMS subject classifications. 68Q25, 68W25, 90B18, 90B20, 91A10

DOI. 10.1137/090769600

1. Introduction.

1.1. Selfish routing and the price of anarchy.

An important research goal is to understand when the equilibria of a noncooperative game approximate the ideal outcome that would be implemented by an all-powerful and altruistic designer. The most popular approximation measure used for this purpose is the *price of anarchy* of a game under a given objective function, which is defined as the worst-case ratio between the objective function value of a Nash equilibrium of the game and that of an optimal solution [20, 24].

This paper studies the price of anarchy of *selfish routing*, a mathematical model defined by Wardrop [32] to describe how noncooperative agents route traffic in a network with congestion. Formally, the game takes place in a directed multicommodity

*Received by the editors September 1, 2009; accepted for publication (in revised form) August 30, 2011; published electronically December 6, 2011. Preliminary versions of these results appeared in *Proceedings of the* 15*th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2004, pp. 333–334, 973–974 and in *Proceedings of the* 32*nd International Colloquium on Automata, Languages, and Programming*, 2005, pp. 497–512.

http://www.siam.org/journals/sidma/25-4/76960.html

†Center for Bioinformatics and Computational Biology, University of Maryland, College Park, MD 20742 (henrylin@umiacs.umd.edu). This author's research was done at UC Berkeley and was supported by a UC Berkeley Research Fellowship.

‡Department of Computer Science, Stanford University, Stanford, CA 94305 (tim@cs.stanford.edu). This author's research was supported in part by an NSF Postdoctoral Fellowship, ONR grant N00014-04-1-0725, and DARPA grant W911NF-04-9-0001.

§Department of Computer Science, Cornell University, Ithaca, NY 14853 (eva@cs.cornell.edu). This author's research was supported in part by NSF grant CCR-032553, ITR grant 0311333, and ONR grant N00014-98-1-0589.

¶Google Inc., Mountain View, CA 94043 (walkover@gmail.com).

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 small independent network users, so that each individual has negligible impact on the experience of others. Each user seeks a minimum latency path, given the congestion imposed by the rest of the traffic. Under this assumption, 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. For the average latency incurred by traffic, the price of anarchy of selfish routing is well understood, and exact worst-case bounds are known under a wide variety of assumptions [7, 10, 12, 13, 25, 27, 30].

The first goal of this paper is to study the price of anarchy of selfish routing with respect to the *maximum* latency incurred by a user. In a flow at Nash equilibrium in a single-commodity network, all flow is routed along shortest paths and incurs the same latency. In contrast, the flow minimizing the average latency can be "unfair," in that some users may need to be sacrificed to very costly paths to reduce the congestion incurred by others [26]. For this reason, it can be more appropriate to compare a flow at Nash equilibrium to the flow minimizing the maximum latency. In multicommodity networks, even the flow at Nash equilibrium can be unfair. The best way to compare different flows is not clear, as different users will have different preferences. The two most standard objective functions are the average latency and the min-max objective function considered here. See also Correa, Schulz, and Stier Moses [11] for further discussion and motivation.

The price of anarchy for the maximum latency objective function turns out to be closely related to *Braess's paradox*, which is the following counterintuitive phenomenon: removing edges from a selfish routing network can *decrease* the latency incurred by all of the traffic at equilibrium. Suppose there is a network in which edge removals can decrease the latency of all traffic by an α factor. Then, α is also a lower bound on the worst-case price of anarchy for the maximum latency objective: the equilibrium flow in this network after the edges have been removed is a feasible flow in the original network with maximum latency at least an α factor smaller than the flow at Nash equilibrium. Equivalently, an upper bound on the price of anarchy for the maximum latency objective upper bounds the largest-possible decrease in latency that can be effected with edge removals. The second goal of this paper is to prove new upper and lower bounds on the worst-case severity of Braess's paradox.

1.2. Our results: Single-commodity networks. In analyzing the worst-case severity of Braess's paradox in single-commodity networks, Roughgarden [29] proved that, for every $n \geq 2$, removing $\lfloor n/2 \rfloor - 1$ edges from a single-commodity network with n vertices can decrease the common latency experienced by all of the traffic by a factor of $\lfloor n/2 \rfloor$. Here, we prove that this construction is optimal in the following sense: for every integer $c \geq 1$, the only way to decrease the latency experienced by traffic by a factor strictly greater than c is to remove at least c edges from the network. In particular, for a single edge removal, which we prove can only decrease the latency of all traffic by a factor of 2, *no network is worse than Braess's original four-node network.* Along the way, we give the first combinatorial proof of a useful monotonicity property of flows at Nash equilibrium.

The construction in [29] immediately implies a lower bound of $\lfloor n/2 \rfloor$ on the worst-case price of anarchy with the maximum latency objective in n-vertex single-commodity networks. We devise a different single-commodity construction that gives a lower bound of $n-1$, and we prove a matching upper bound (for every $n \geq 2$) on the price of anarchy in single-commodity networks with arbitrary continuous and nondecreasing latency functions. In contrast, no finite upper bound exists for the price of anarchy with the *average* latency objective and arbitrary latency functions, even in networks with only two nodes and two links [30].

1.3. Our results: Multicommodity networks. The rest of our results concern the more challenging setting of multicommodity networks. We establish exponential 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, demonstrating in the process that both can be much larger in multicommodity networks than in single-commodity ones. Our two primary results are the following:

- We give a construction, based on the Fibonacci numbers, that shows that removing one edge from a two-commodity network with n vertices can decrease 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 that proves the first result implies that for all existing approximation-type analyses of selfish routing that were known to hold only in single-commodity networks—in [29] and in this paper—there cannot be any reasonable extension to multicommodity networks, even those with only two commodities. This dichotomy between single- and two-commodity networks stands in contrast to the provably irrelevant role that the number of commodities plays in the price of anarchy for the average latency objective [10, 27].

The aforementioned connection between the price of anarchy for the maximum latency objective and Braess's paradox implies that our upper and lower bounds on both the price of anarchy and on the worst-possible severity of Braess's paradox are close to tight for networks with a constant number of commodities.

Finally, we consider the problem 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 [29] for the single-commodity version of the problem, we prove that there is no polynomial-time algorithm for this network design problem with a subexponential approximation ratio (assuming $P \neq NP$). Since our upper bound on the price of anarchy trivially implies that an exponential approximation ratio is achievable, this network design problem is a rare example of a natural optimization problem with intrinsically exponential approximability.

1.4. Related work. The maximum latency objective has been extensively studied in a game-theoretic scheduling context, which corresponds to a network of parallel links and players that control a nonnegligible amount of traffic. Koutsoupias and Papadimitriou [20] initiated this line of research, and Vöcking in [23, Chapter 20] surveys it. Weitz [33] was the first to study the price of anarchy of selfish routing under the maximum latency objective and noted that, for single-commodity networks, the price of anarchy for the maximum latency is no more than that for the average latency objective. Weitz [33] also proved that the price of anarchy for the maximum latency objective is at least approximately $n/2$ in multicommodity networks with linear latency functions and a large number of commodities; for an analogous single-sink example, see [11]. Concurrently with the conference version of some of our work [28], Correa, Schulz, and Stier Moses [11] studied various fairness objective functions including the maximum latency objective. The results of [11] mostly concern the computational complexity of computing an optimal solution and the extent to which multiple objective functions can be simultaneously optimized for restricted classes of latency functions; they are disjoint from those presented in this paper. Other studies of fairness issues in selfish routing networks include [6, 16, 26]. For work on computing or approximating a flow with minimum maximum latency in polynomial time, see [11, 19] and the references therein.

Braess's paradox was first presented in [4] and has motivated a vast number of follow-up papers; see [29] for an overview. The worst-case severity of Braess's paradox was first studied in [18] for Braess's four-node network topology and independently in [29] for general single-commodity networks.

Finally, subsequent to the conference versions of the present work [21, 22, 28], several papers have given upper and lower bounds on the worst-case severity of Braess's paradox and the price of anarchy with the maximum latency objective in *atomic* selfish routing networks, where there is a finite number of players who each control a nonnegligible amount of traffic [1, 2, 5, 8, 14].

1.5. Organization. Section 2 formally defines selfish routing networks and their equilibria, the price of anarchy, and our measure of the severity of Braess's paradox. Sections 3 and 4 consider bounds on the price of anarchy and Braess's paradox, respectively, in single-commodity networks. Sections 5 and 6 prove lower and upper bounds, respectively, on the worst-case severity of Braess's paradox and the price of anarchy in multicommodity networks. Section 7 describes a network design problem, motivated by detecting Braess's paradox in multicommodity networks, for which the best-possible approximation ratio of a polynomial-time algorithm is exponential in the network size (assuming $P \neq NP$).

2. Preliminaries.

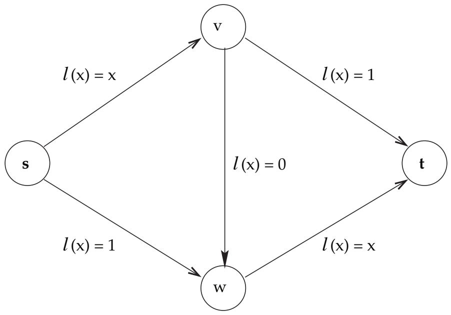
2.1. The model. We study the standard model of selfish routing, with a multicommodity flow network described by a directed graph $G = (V, E)$ and k source-destination vertex pairs $(s_1, t_1), \ldots, (s_k, t_k)$. We denote by r_i the amount of traffic that wishes to travel from the source s_i to the destination t_i—the *traffic rate*. *Single-commodity networks* are those with $k = 1$. The graph G can contain parallel edges, but we can exclude self-loops. We denote the s_i-t_i paths of G by $\mathcal{P}_i$ and assume that $\mathcal{P}_i$ is nonempty for all i. We use $\mathcal{P}$ to denote $\cup_{i=1}^{k} \mathcal{P}_i$.

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

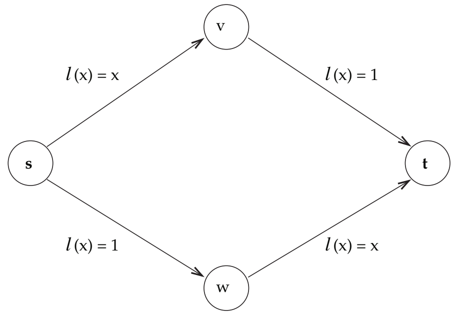
To model congestion effects, we give each edge e a nonnegative, continuous, nondecreasing *latency function* ℓ_e describing the time needed to traverse the edge as a function of the edge congestion f_e. Given a flow f, the latency ℓ_P of a path P is the sum of the latencies of the edges in the path: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

2.2. Flows at Nash equilibrium. Assuming that all network users have negligible size and want to minimize the latency experienced, we expect all users to travel on paths with the minimum-possible latency. We formalize this in the definition of a flow at Nash equilibrium.

DEFINITION 2.1 (flow at Nash equilibrium [32]). *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 $f_{P_1} > 0$, $\ell_{P_1}(f) \leq \ell_{P_2}(f)$.*



(a) Braess's paradox



(b) After an edge removal

FIG. 1. *Example 2.2. One unit of selfish traffic travels from s to t. Edges are labeled with their latency functions. In* (a), *the flow at Nash equilibrium sends all traffic on the path $s \to v \to w \to t$, and the common latency is* 2. *In* (b), *the flow at Nash equilibrium splits the traffic between the paths $s \to v \to t$ and $s \to w \to t$, and the common latency is* 3/2.

Example 2.2 (Braess's paradox [4]). Consider the selfish routing network shown in Figure 1(a), with one unit of traffic traveling from s to t. In the unique flow at Nash equilibrium, all traffic uses the path $s \to v \to w \to t$ and incurs 2 units of latency. In the unique flow at Nash equilibrium in the network in Figure 1(b), the traffic is split evenly between the two s-t paths and all traffic experiences 3/2 units of latency. Thus removing the edge (v, w) from the first network decreases the common latency of traffic in a Nash flow by a 4/3 factor.

If we change the latency functions of edges (s, v) and (w, t) from $\ell(x) = x$ to $\ell(x) = x^d$, then removing the edge (v, w) decreases the latency of traffic by a factor that approaches 2 as $d \to \infty$.

Under our assumptions that latency functions are continuous and nondecreasing, Nash flows always exist and all Nash flows of an instance induce the same latency on every edge.

PROPOSITION 2.3 (existence and uniqueness of Nash flows [3]). *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 $\ell_e(f_e) = \ell_e(\tilde{f}_e)$ for every edge e.*

A stronger form of Proposition 2.3(a) holds in single-commodity networks.

PROPOSITION 2.4 (existence of an acyclic Nash flow [29]). *In every single-commodity instance, there is a Nash flow f such that the subgraph of edges with $f_e > 0$ is directed acyclic.*

We use Proposition 2.4 in the proofs of Theorems 3.2 and 4.1.

In several proofs (of Theorems 3.2, 4.1, and 6.4), we use the following straightforward characterization of the flows at Nash equilibrium of an instance. It expresses the shortest-path condition of Definition 2.1 in terms of shortest-path distance labels.

PROPOSITION 2.5 (Nash flows via shortest-path labels [29]). *Let f be a flow feasible for the instance (G, r, ℓ). For a vertex v in G and a commodity i, let $d^i(v)$ denote the length, with respect to edge lengths $\ell_e(f_e)$, of a shortest s_i-v path in G. Then f is at Nash equilibrium if and only if for every pair v, w of vertices in G, every commodity i, and every v-w path P the following hold:*

(a) $d^i(w) - d^i(v) \le \sum_{e \in P} \ell_e(f_e)$; *and*

(b) *if $f_e^{(i)} > 0$ for every edge $e \in P$, then $d^i(w) - d^i(v) = \sum_{e \in P} \ell_e(f_e)$.*

For example, if f denotes the flow at Nash equilibrium in the network in Figure 1(a), then the corresponding distance labels are $d(s) = 0$, $d(v) = d(w) = 1$, and $d(t) = 2$.

2.3. The price of anarchy. The price of anarchy of a game is defined with respect to an objective function. In this paper, we consider the maximum latency $M(f)$ incurred by a flow f:

$$M(f) = \max_{P \in \mathcal{P}\,:\, f_P > 0} \ell_P(f).$$

With respect to an instance (G, r, ℓ), a flow that minimizes $M(\cdot)$ over all feasible flows is called *optimal*. Since the feasible flows of an instance form a compact subset of a Euclidean space and $M(\cdot)$ is a continuous function, every instance admits an optimal flow.

The price of anarchy of a selfish routing network is the ratio of the objective function values of a flow at Nash equilibrium and an optimal flow.

DEFINITION 2.6 (price of anarchy). *If (G, r, ℓ) is an instance, then the* price of anarchy of (G, r, ℓ)*, denoted by $\rho(G, r, \ell)$, is the ratio $M(f)/M(f^*)$, where f is a Nash flow and f^* is an optimal flow.*

Proposition 2.3 implies that all Nash flows of an instance have the same maximum latency, and so 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 to be 1. For example, in the first network of Example 2.2, shown in Figure 1(a), the price of anarchy is 4/3: $M(f) = 2$ for the Nash flow f, while $M(f^*) = 3/2$ for the optimal flow f^*, which splits traffic equally between the paths $s \to v \to t$ and $s \to w \to t$.

2.4. The Braess ratio. A quantity related to but different from the price of anarchy is the *Braess ratio* of a selfish routing network, defined as the largest factor by which the equilibrium latency of all traffic can be decreased by edge removals. To define this quantity formally, we use the notation $L_i(G, r, \ell)$ to denote the common latency of the ith commodity's traffic in a Nash flow for the instance (G, r, ℓ); by Proposition 2.3(b), this number is well defined (i.e., independent of the particular Nash flow).

DEFINITION 2.7 (Braess ratio). *The* Braess ratio $\beta(G, r, \ell)$ *of an instance* (G, r, ℓ) *is*

$$\beta(G, r, \ell) = \max_{H \subseteq G} \min_{i=1}^{k} \frac{L_i(G, r, \ell)}{L_i(H, r, \ell)},$$

where H ranges over the subnetworks of G that contain an s_i-t_i path for each i.

In Definition 2.7, we interpret the ratio 0/0 as 1. Observe that the Braess ratio of a multicommodity instance is large only if removing some set of edges decreases the latency incurred by the traffic of every commodity by a large amount. In a single-commodity network, Definition 2.7 simplifies to $\beta(G, r, \ell) = \max_H [L(G, r, \ell)/L(H, r, \ell)]$, where $L(\cdot, \cdot, \cdot)$ denotes the equilibrium latency of all traffic in such a network. For example, in the first network of Example 2.2, the Braess ratio is 4/3.

An upper bound on the price of anarchy of an instance applies immediately to its Braess ratio.

PROPOSITION 2.8 (price of anarchy upper bounds Braess ratio). *For every instance (G, r, ℓ), the Braess ratio $\beta(G, r, \ell)$ is at most the price of anarchy $\rho(G, r, \ell)$.*

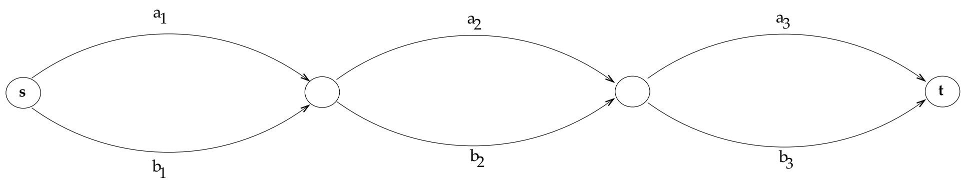


FIG. 2. *Example* 3.1 *with* $n = 4$. *The worst-case price of anarchy with the maximum latency objective in single-commodity networks with* n *vertices is at least* $n - 1$.

Proof. For every subgraph H of G, a flow at Nash equilibrium f^H for (H, r, ℓ) is a feasible flow for (G, r, ℓ). By the definition of the price of anarchy, the maximum latency of a flow at Nash equilibrium for (G, r, ℓ) is at most a $\rho(G, r, \ell)$ factor larger than that of f^H. Taking i^* as a commodity maximizing $L_i(H, r, \ell)$, we have $L_{i^*}(G, r, \ell) \leq \rho(G, r, \ell) \cdot L_{i^*}(H, r, \ell)$, and the proof is complete. □

3. The maximum latency in single-commodity networks. To begin, we give matching upper and lower bounds on the worst-case price of anarchy for the maximum latency objective in single-commodity networks. We start with the lower bound.

Example 3.1 (lower bound on the price of anarchy in single-commodity networks). Let $n \geq 2$ be an integer. Let G be the network with vertices $v_1, \ldots, v_n$, with $s = v_1$ and $t = v_n$, and with two edges, a_i and b_i, directed from v_i to v_{i+1} for each $i = 1, \ldots, n-1$. See Figure 2. For each such i, edge a_i is given the constant latency function $\ell_{a_i}(x) = 1$ and edge b_i a latency function that satisfies $\ell_{b_i}((n-2)/(n-1)) = 0$ and $\ell_{b_i}(1) = 1$. The flow f that routes one unit of flow on edge b_i for all i is at Nash equilibrium for $(G, 1, \ell)$ with $M(f) = n - 1$. On the other hand, the flow f^* that splits traffic evenly between the $n - 1$ paths that eschew exactly one edge of the form b_i is feasible for $(G, 1, \ell)$, with $M(f^*) = 1$.

Example 3.1 shows that the worst-case price of anarchy in single-commodity instances with n vertices is at least $n - 1$. This improves over the lower bound of $\lfloor n/2 \rfloor$ that follows from Proposition 2.8 and the lower bound on the Braess ratio in [29]. The next theorem provides a matching upper bound.

THEOREM 3.2 (upper bound on the price of anarchy in single-commodity networks). *For every* $n \geq 2$ *and every single-commodity instance* (G, r, ℓ) *with* n *vertices,* $\rho(G, r, \ell) \leq n - 1$.

Proof. Fix $n \geq 2$ and let (G, r, ℓ) be an n-vertex single-commodity instance. Let f be a Nash flow for (G, r, ℓ), f^* a feasible flow for (G, r, ℓ), and $d(v)$ the shortest-path distance from s to v with respect to edge lengths $\ell_e(f_e)$, as in Proposition 2.5. That proposition implies that $M(f) = d(t)$.

Since all Nash flows have equal maximum latency, Proposition 2.4 implies that we can assume without loss of generality that f is a directed acyclic flow. This implies that the vertices of G can be sorted $s = v_1, v_2, \ldots, v_q = t, v_{q+1}, \ldots, v_n$ in topological order with respect to f such that $d(v)$ is nondecreasing in the ordering. To see why, start from an arbitrary topological ordering and repeatedly swap consecutive vertices v, w for which $d(v) > d(w)$. Proposition 2.5(b) implies that every such swap yields another topological ordering of the vertices with respect to f, and after a finite number of swaps the desired ordering is obtained.

We now pick consecutive vertices v, w that precede or equal t in the ordering and that maximize the difference $d(w) - d(v)$; since $d(t)$ is the sum of $q - 1 \leq n - 1$ such differences, the maximum difference is at least $d(t)/(n-1)$. Let S denote the set of vertices between s and v, inclusive, in the ordering. The set S is an s-t cut, and since vertices are sorted topologically with respect to f, no f-flow enters S and hence the amount of f-flow exiting S is precisely r. The s-t flow f^* sends at least r units of flow out of the cut S, so there is an edge $e = (u, x)$ exiting S on which $f_e^* \geq f_e$ and $f_e^* > 0$. Hence, $M(f^*) \geq \ell_e(f_e^*) \geq \ell_e(f_e) \geq d(x) - d(u)$, where the final inequality follows from Proposition 2.5(a). Since u is or precedes v in the ordering, x is or succeeds w in the ordering, and d-values can only increase with the ordering, $d(x) - d(u) \geq d(w) - d(v)$. Thus,

$$M(f^*) \geq \ell_e(f_e) \geq d(w) - d(v) \geq \frac{d(t)}{n-1},$$

and the proof is complete. ∎

4. A monotonicity result and Braess's paradox bounds in single-commodity networks. This section gives the first bound on the Braess ratio that is parameterized by the number of removed edges. Along the way, we give the first combinatorial proof of an important monotonicity result for Nash flows in single-commodity selfish routing networks.

Our most general result is that the size of the largest matching of $V \setminus \{s, t\}$ among the removed edges controls how much the latency of a Nash flow can decrease.

THEOREM 4.1 (parameterized upper bound on Braess's paradox). *Let (G, r, ℓ) be a single-commodity instance, with $G = (V, E)$. Let H be a subgraph of G and S be the edges that are in G but not H. If every subset of S that forms a matching of $V \setminus \{s, t\}$ has size as most c, then*

$$L(G, r, \ell) \leq (c + 1) \cdot L(H, r, \ell).$$

We immediately obtain the promised upper bound that is parameterized by the number of removed edges. This upper bound was first conjectured by Kameda [17].

COROLLARY 4.2 (bounding Braess's paradox with limited edge removals). *If (G, r, ℓ) is a single-commodity instance and H is obtained from G by removing at most c edges, then*

$$L(G, r, \ell) \leq (c + 1) \cdot L(H, r, \ell).$$

Theorem 4.1 also gives a new proof of one of the main results in [29].

COROLLARY 4.3 (bounding Braess's paradox with unlimited edge removals [29]). *If (G, r, ℓ) is a single-commodity instance with n vertices, then*

$$\beta(G, r, \ell) \leq \left\lfloor \frac{n}{2} \right\rfloor.$$

Proof. Since there are only $n - 2$ nodes in G that are not s or t, every matching of $V \setminus \{s, t\}$ contains at most $\lfloor (n-2)/2 \rfloor = \lfloor n/2 \rfloor - 1$ edges. Theorem 4.1 now implies the corollary. ∎

A family of networks described in [29] shows that Theorem 4.1 and Corollaries 4.2 and 4.3 are tight in the worst case for all values of $c \geq 1$ and $n \geq 2$.

We now turn toward proving Theorem 4.1. We begin with a definition.

DEFINITION 4.4 (light and heavy edges; alternating paths). *Let f and $\tilde{f}$ be flows feasible for the instances (G, r, ℓ) and $(G, \tilde{r}, \ell)$, respectively:*

(a) *An edge e of G is* $(f,\tilde{f})$-light *if* $f_e \le \tilde{f}_e$ *and* $\tilde{f}_e > 0$, $(f,\tilde{f})$-heavy *if* $f_e > \tilde{f}_e$, *and* $(f,\tilde{f})$-null *if* $f_e = \tilde{f}_e = 0$.

(b) *An undirected path is* $(f,\tilde{f})$-alternating *if it comprises only forward light edges and backward heavy edges.*

When the context is clear, we drop the dependence on f and $\tilde{f}$ for the terms in Definition 4.4.

Example 4.5 (an alternating path). Let f be the Nash flow and $\tilde{f}$ be the optimal flow in Figure 1(a). Then, edges (s,v), (v,w), and (w,t) are $(f,\tilde{f})$-heavy while edges (s,w) and (v,t) are $(f,\tilde{f})$-light. The unique $(f,\tilde{f})$-alternating s-t path is $s \to w \to v \to t$.

We now prove that an s-t alternating path exists when comparing one flow to another at the same or an increased traffic rate.

LEMMA 4.6 (existence of alternating paths). *Let f and $\tilde{f}$ be flows feasible for the single-commodity instances (G,r,ℓ) and $(G,\tilde{r},\ell)$, respectively, with $r \le \tilde{r}$. Then, there is an $(f,\tilde{f})$-alternating s-t path. Moreover, if f is directed acyclic, then every such path begins and ends with an $(f,\tilde{f})$-light edge.*

Proof. Suppose for contradiction that there is no $(f,\tilde{f})$-alternating s-t path, and let S denote the set of nodes reachable from s via such paths. The set S contains s and, by assumption, does not contain t; it is therefore an s-t cut. Since the same net amount of s-t flow crosses every s-t cut of a flow network, we have

$$\sum_{e\in\delta^+(S)} f_e - \sum_{e\in\delta^-(S)} f_e = r \tag{1}$$

and

$$\sum_{e\in\delta^+(S)} \tilde{f}_e - \sum_{e\in\delta^-(S)} \tilde{f}_e = \tilde{r}, \tag{2}$$

where $\delta^+(S)$ is the set of edges exiting S and $\delta^-(S)$ is the set of edges entering S.

Since vertices in S can be reached from s via $(f,\tilde{f})$-alternating paths and vertices outside S cannot, edges that exit S are heavy or null. In addition, one such edge is heavy, since otherwise the left-hand side of (1) would be nonpositive. Similarly, edges that enter S are light or null. These observations imply that the left-hand side of (1) is strictly greater than that of (2), contradicting the fact that $r \le \tilde{r}$.

Moreover, if f is directed acyclic, then it sends no flow into s or out of t. Thus, the first and last edges of every $(f,\tilde{f})$-alternating s-t path are light. ∎

To prepare for the proof of Theorem 4.1, we first prove an orthogonal monotonicity result that is interesting in its own right. An intuitive but far from obvious fact is that the latency encountered by traffic in a Nash flow can only increase as we inject new traffic into the system (recall that edge latency functions are nondecreasing). It was first proved by Hall [15], in a more general multicommodity network setting, using techniques for sensitivity analysis of convex programs. Here, we give a direct and combinatorial proof. The proof techniques will be directly useful in our proof of Theorem 4.1.

THEOREM 4.7 (equilibrium latency is nondecreasing in the traffic rate [15]). *For every network G with one source-destination pair and latency functions ℓ, the value $L(G,r,\ell)$ is nondecreasing in r.*

Proof. Let f and $\tilde{f}$ be Nash flows for (G,r,ℓ) and $(G,\tilde{r},\ell)$, respectively, with $r \le \tilde{r}$. For a vertex v, let $d(v)$ and $\tilde{d}(v)$ denote the shortest-path distances from s to v with respect to edge lengths $\ell_e(f_e)$ and $\ell_e(\tilde{f}_e)$, respectively. These are the same shortest-path distance labels as in Proposition 2.5. By definition, $L(G, r, \ell) = d(t)$ and $L(G, \tilde{r}, \ell) = \tilde{d}(t)$. The proposition asserts that $d(t) \le \tilde{d}(t)$.

We prove the stronger result that $d(v) \le \tilde{d}(v)$ for all vertices v of an arbitrary $(f, \tilde{f})$-alternating s-t path P. At least one such path exists by Lemma 4.6. We proceed by induction. For the base case, $d(s) = \tilde{d}(s) = 0$. Suppose that $d(v) \le \tilde{d}(v)$ for some vertex v on P, and let w be the next vertex on P. There are now two cases.

First, suppose that edge $e = (v, w)$ is $(f, \tilde{f})$-light. Then, $\ell_e(f_e) \le \ell_e(\tilde{f}_e)$ and $\tilde{f}_e > 0$. Since f and $\tilde{f}$ are Nash flows, Proposition 2.5 and the inductive hypothesis imply that

$$d(w) \le d(v) + \ell_e(f_e) \le \tilde{d}(v) + \ell_e(\tilde{f}_e) = \tilde{d}(w).$$

Now suppose that edge $e = (w, v)$ is $(f, \tilde{f})$-heavy. Since $f_e > 0$ and f is a Nash flow, Proposition 2.5 implies that

$$d(v) = d(w) + \ell_e(f_e). \tag{3}$$

Similarly, for $\tilde{f}$ we have

$$\tilde{d}(v) \le \tilde{d}(w) + \ell_e(\tilde{f}_e). \tag{4}$$

Since $d(v) \le \tilde{d}(v)$ by the inductive hypothesis, and $\ell_e(\tilde{f}_e) \le \ell_e(f_e)$ since e is $(f, \tilde{f})$-heavy, equation (3) and inequality (4) are compatible only if $d(w) \le \tilde{d}(w)$. This completes the inductive step and the proof of the theorem. ∎

Finally, we show how a refinement of the proof of Theorem 4.7 proves Theorem 4.1.

Proof of Theorem 4.1. Let f and $\tilde{f}$ be an acyclic Nash flow and a Nash flow for (G, r, ℓ) and (H, r, ℓ), respectively; the former exists by Proposition 2.4. We view $\tilde{f}$ as a flow in the larger network G in the obvious way. As in the proof of Theorem 4.7, let d and $\tilde{d}$ denote shortest-path distances with respect to the edge latencies induced by f and $\tilde{f}$ in G and H, respectively. This proof must differ from that of Theorem 4.7, as $\tilde{f}$ is a Nash flow in H but not in G.

Let P be an $(f, \tilde{f})$-alternating s-t path, which exists by Lemma 4.6. A *segment* of P is a maximal subpath of P that contains only $(f, \tilde{f})$-light or only $(f, \tilde{f})$-heavy edges. Edges that are in G but not in H are called *absent*. Since $\tilde{f}_e > 0$ on $(f, \tilde{f})$-light edges, absent edges can only reside in $(f, \tilde{f})$-heavy segments. The key claim is that if v is a vertex at the end of a segment of P and c (heavy) segments of P between s and v contain an absent edge, then

$$d(v) \le \tilde{d}(v) + c \cdot \tilde{d}(t). \tag{5}$$

This claim implies the theorem. To see why, first apply (5) to t to obtain

$$d(t) \le \tilde{d}(t) + c \cdot \tilde{d}(t) = (c + 1) \cdot \tilde{d}(t),$$

where c is the number of segments of P that include an absent edge. This inequality reduces the proof of the theorem to exhibiting c absent edges that form a matching of $V \setminus \{s, t\}$. By Definition 4.4 and Lemma 4.6, $(f, \tilde{f})$-heavy segments of P are disjoint from each other and from s and t. Picking one absent edge from each $(f, \tilde{f})$-heavy segment that contains one provides the desired matching.

We now prove (5) by induction on the segments of P. The inequality trivially holds when $v = s$, so suppose it holds for a vertex v that is last on a segment of P or is equal to s. We wish to prove (5) for w, defined as the last vertex on the next segment. If no edges in the segment between v and w are absent, then (5) holds by the arguments in the penultimate or the final paragraph of the proof of Theorem 4.7, depending on whether the segment contains light or heavy edges, respectively.

Since absent edges can only be heavy, we can finish the proof by establishing the inductive hypothesis when the segment between v and w comprises heavy backward edges, at least one of which is absent. First, since $f_e > 0$ on all of these edges, Proposition 2.5(b) implies that $d(w) \leq d(v)$. Since the path P begins with a light edge (Lemma 4.6), $v \neq s$ and there is a light edge entering v. Since $\tilde{f}$ routes flow into v, it must route flow from v to t. By Proposition 2.5(b), $\tilde{d}(v) \leq \tilde{d}(t)$. Combining what we know with the inductive hypothesis completes the proof:

$$d(w) \leq d(v) \leq \tilde{d}(v) + c \cdot \tilde{d}(t) \leq (c+1) \cdot \tilde{d}(t) \leq \tilde{d}(w) + (c+1) \cdot \tilde{d}(t). \qquad \square$$

5. Braess's paradox in multicommodity networks. This section proves that there is a "phase transition" in the worst-case severity of Braess's paradox between single-commodity networks, where the Braess ratio is always polynomial in the network size, and multicommodity networks, where the Braess ratio can be exponential in the network size, even with only two commodities. This construction is also the starting point for our inapproximability results in section 7.

Our family of two-commodity instances is closely related to the *Fibonacci numbers*, where the pth Fibonacci number F_p is defined as $F_0 = 0$, $F_1 = 1$, 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. Our main result in this section is the following.

THEOREM 5.1 (exponential lower bound in two-commodity networks). *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 every 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 5.1 has a number of implications. We first note two immediate corollaries.

COROLLARY 5.2. *Removing a single edge from an n-vertex two-commodity instance can decrease the latency of all traffic by a $2^{\Omega(n)}$ factor as $n \to \infty$.*

COROLLARY 5.3. *The worst-case price of anarchy in two-commodity instances with at most n vertices is $2^{\Omega(n)}$ as $n \to \infty$.*

These corollaries obviously apply also to networks with more than two commodities.

Theorem 5.1 and Corollaries 5.2 and 5.3 show that Theorem 3.2 and Corollaries 4.2 and 4.3 utterly fail to extend to multicommodity networks. This dichotomy stands in contrast to bounds on the price of anarchy for the average latency objective function, where there is provably no separation between single-commodity and multicommodity instances [27].

We now give the construction of the family of instances claimed in Theorem 5.1. We begin by defining the graph G^p for $p \geq 1$; see Figure 3. We describe the construction only for odd p; the construction for even p is almost the same. 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 3, 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 3, 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

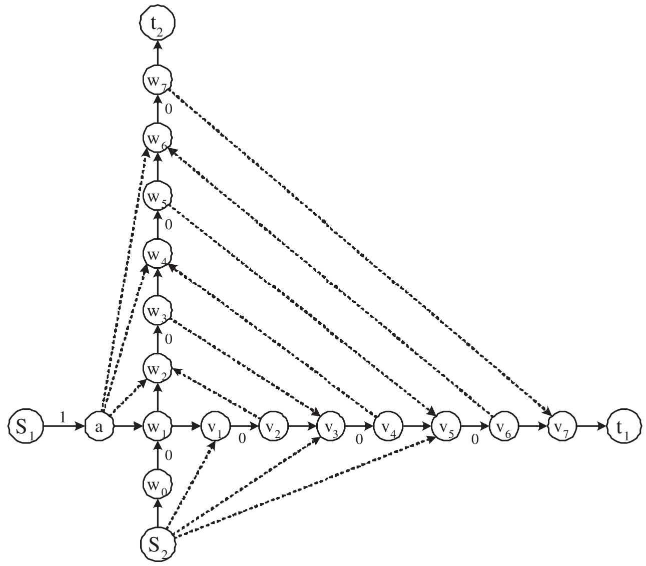


FIG. 3. *The instance (G^p, r^p, ℓ^p) when $p = 7$ without the "extra edge" (s_1, w_0). Solid edges carry flow at Nash equilibrium; dotted edges do not. Edge latencies are with respect to the Nash flow. Unlabeled edges have zero latency.*

positive even i; (s_2, v_i) for all odd i at most $p - 2$; and (w_i, v_i) for all odd i. Finally, we complete G^p by adding what we call an *extra edge*, defined as the edge (s_1, w_0).

For all p, the traffic rates are $r_1^p = r_2^p = 1$. To complete the construction, we need to describe the edge latency functions. All edges possess either a constant latency function, or a latency function that approximates a step function. Precisely, for a positive integer i and a positive real number δ, g_δ^i denotes a continuous, nondecreasing function with $g_\delta^i(x) = 0$ for $x \leq 1$ and $g_\delta^i(x)$ equal to the ith Fibonacci number F_i for $x \geq 1 + \delta$. (The function g_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. The latency functions ℓ^p for G^p are, for some sufficiently small δ as follows: for each $i > 0$, edge e_i receives the latency function $\ell^p(x) = g_\delta^i(x)$; edge e_0 receives the latency function $\ell^p(x) = g_\delta^1(x)$; edge (s_1, a) receives the latency function $\ell^p(x) = 1$; and all other edges receive the latency function $\ell^p(x) = 0$.

We now prove Theorem 5.1 for odd p; the arguments for even p are almost identical. 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 on both P_1 and P_2, then f is at Nash equilibrium for (H^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 3.

To finish the proof of Theorem 5.1 (for p odd), we prove part (b) via a sequence of lemmas. The first one requires some definitions. We 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 $i \in \{0, 1, \ldots, p - 1\}$. If f floods (G^p, r^p, ℓ^p), then all edge latencies are at their maximum, as in Figure 4. Next, for i even and positive, Q_i denotes the s_1-t_1 path that uses edge e_i as a "short cut": $s_1 \to a \to w_i \to w_{i+1} \to v_{i+1} \to \cdots \to v_p \to t_1$. The path Q_0 has the same form, except vertex a is skipped via the extra edge: $s_1 \to w_0 \to w_1 \to v_1 \to \cdots \to v_p \to s_1$.

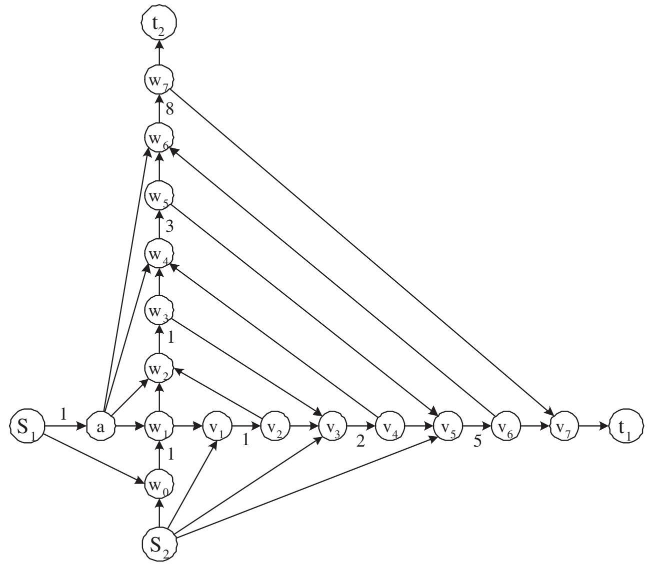


FIG. 4. *Nash flow in* (G^p, r^p, ℓ^p)*, with* $p = 7$*. All edges carry flow. Edge latencies are with respect to the Nash flow. Unlabeled edges have zero latency.*

For i odd, Q_i denotes the s_2-t_2 path that uses edge e_i as a "short cut": $s_2 \to v_i \to v_{i+1} \to w_{i+1} \to \cdots \to w_p \to t_2$. The paths $Q_0, \ldots, Q_{p-1}$, together P_1 and P_2, are *short paths*. The next lemma justifies this terminology, at least for flows that flood the instance (G^p, r^p, ℓ^p).

LEMMA 5.4. *If* f *floods* (G^p, r^p, ℓ^p) *with* p *odd, then the following hold:*

(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 prove only part (b) of Lemma 5.4, as the proof of part (a) is similar. In the proof, we use the following lemma about the Fibonacci numbers, which is easy to verify by induction.

LEMMA 5.5. *Let* j *and* p *be odd positive integers with* $j < p$ *and* I *be the even numbers between* j *and* p*. Then,* $F_j + \sum_{i \in I} F_i = F_p$*.*

Proof of Lemma 5.4. 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 can assume that $j > 0$, since the $j = 0$ and $j = 1$ cases are the same. Since $j > 0$ is maximal, P contains e_j and also e_i for all even i that are strictly between j and p. Since f floods (G^p, r^p, ℓ^p), Lemma 5.5 implies that $\ell_P(f) \geq F_p$. Moreover, this inequality holds with equality for short paths. ∎

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 nonconstant latency functions of (G^p, r^p, ℓ^p) increase as the amount of flow on the edge exceeds 1.

LEMMA 5.6. *For all* p *odd and* δ *sufficiently small, there is flow* f*, with* $f_P > 0$ *only for short paths* P*, which 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 show that $f_{e_i} \geq 1 + \delta$ for every

$i \in \{0, 1, \ldots, p-1\}$, provided δ is sufficiently small. We prove this inequality only for odd i; the argument for even i is very 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 more than

$$2^{-(i+1)} + \left(1 - \sum_{j=0}^{\infty} 2^{-(i+2+2j)}\right) = 1 + 2^{-(i+1)} - \frac{4}{3} \cdot 2^{-(i+2)} > 1 + 2^{-(i+3)}. \tag{6}$$

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

Theorem 5.1(b) now follows immediately from Definition 2.1, Lemma 5.4, and Lemma 5.6.

6. Upper bounds on the price of anarchy in multicommodity networks. This section proves upper bounds on the price of anarchy and, as a consequence, on the worst-possible severity of Braess's paradox. Our upper bound comes close to the lower bound of Theorem 5.1.

We begin with a weak bound on the price of anarchy that depends on parameters other than the network size. While not interesting in its own right, it plays an important role in later proofs in this section.

LEMMA 6.1. *Let f be a Nash flow and f^* be a feasible flow for an instance (G, r, ℓ), where G has m edges. For every edge $\hat{e}$ of G with $f_{\hat{e}} > f^*_{\hat{e}}$,*

$$\ell_{\hat{e}}(f_{\hat{e}}) \leq \frac{m \sum_{i=1}^{k} r_i}{f_{\hat{e}} - f^*_{\hat{e}}} \cdot \max_{e \in E} \ell_e(f^*_e). \tag{7}$$

Proof. Let $F \subseteq E$ denote the edges e of G for which $f_e > f^*_e$. The definition of a flow at Nash equilibrium easily implies that $\sum_{e \in E} \ell_e(f_e)(f^*_e - f_e) \geq 0$; see also [31]. Proceeding crudely, for $\hat{e} \in F$ we have

$$\ell_{\hat{e}}(f_{\hat{e}})(f_{\hat{e}} - f^*_{\hat{e}}) \leq \sum_{e \in F} \ell_e(f_e)(f_e - f^*_e) \leq \sum_{e \in E \setminus F} \ell_e(f_e)(f^*_e - f_e) \leq \max_{e \in E} \ell_e(f^*_e) \cdot m \cdot \sum_{i=1}^{k} r_i.$$

Rearranging proves the lemma. ◻

We next use Lemma 6.1 as a bootstrap for deriving upper bounds on the price of anarchy that depend only on the network size. We accomplish this as follows. For an arbitrary instance, we 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 define our notion of equivalence so that Lemma 6.1 ensures that the linear program has a bounded maximum and then analyze the vertices of the feasible region of the linear program to derive the following bound.

THEOREM 6.2 (upper bound #1 on the price of anarchy). *If (G, r, ℓ) is an instance with n vertices and m edges, then $\rho(G, r, \ell) = 2^{O(m \log n)}$.*

To implement this proof approach, we need 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 6.3. *Let A be an $m \times n$ matrix with entries in $\{-1, 0, 1\}$ and at most λ nonzero entries in each row. Let b be a real-valued m-vector and let $i \in \{1, 2, \ldots, n\}$.*

If the linear program $\max x_i$ *subject to* $Ax \leq b$ *and* $x \geq 0$ *has a finite maximum, then it is at most* $n\lambda^n \|b\|_\infty$, *where* $\|b\|_\infty$ *denotes* $\max_j |b_j|$.

Proof. Basic linear programming theory ensures that there is an optimal solution x^* to the linear program that is a vertex of the feasible region. (See, e.g., [9].) Since there are n decision variables, the vertex x^* is the unique solution to a square linear system of the form $Cx = z$, where C is an $n \times n$ matrix of constraints (from A and the identity matrix I) and z is the corresponding n-vector of entries from b and of zeros (from the nonnegativity constraints). Using Cramer's rule to compute the solution and expanding determinants, each component of x^* is, at worst, the sum of λ^n nonzero terms, each of magnitude at most $\|b\|_\infty$. The proposition follows. ∎

Proof of Theorem 6.2. 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 first preprocess 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 minimum-possible maximum latency and hence increasing the ρ-value of the instance. Combining these two assumptions, we can assume, without loss of generality, that $\ell_e(f_e^*) \leq M(f^*)$ for every edge e of G: either $f_e^* = 0$ and hence $\ell_e(f_e^*) = 0 \leq M(f^*)$, or $f_e^* > 0$ and hence $\ell_e(f_e^*) \leq M(f^*)$.

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 $\{f_e\}_{e \in E}$ and $\{f_e^*\}_{e \in E}$, as well as the latencies $\{\ell_e(f_e^*)\}_{e \in E}$ with respect to f^*, are held fixed. There is 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 $f_e \leq f_e^*$, $\hat{\ell}_e(f_e) \leq \ell_e(f_e^*)$;
- for all edges e with $f_e \geq f_e^*$, $\hat{\ell}_e(f_e) \geq \ell_e(f_e^*)$.

Additionally, we insist that the (fixed) flow f be at Nash equilibrium with respect to the (variable) latencies $\{\hat{\ell}_e(f_e)\}_{e \in E}$. There are several ways that this requirement can be encoded with linear constraints. For the present proof, we use a naive approach: in our linear program, we insist that

$$\sum_{e \in P} \hat{\ell}_e(f_e) \leq \sum_{e \in \tilde{P}} \hat{\ell}_e(f_e)$$

for every commodity i, and every pair of paths $P, \tilde{P} \in \mathcal{P}_i$ for which $f_e^{(i)} > 0$ for all $e \in P$. Since this linear program has only m variables, we will not be hampered by its potentially massive number of constraints.

By construction, our constraints ensure the following: for every feasible solution $\{\hat{\ell}_e(f_e)\}_{e \in E}$, 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 problem of maximizing the value of a single variable $\hat{\ell}_e(f_e)$ over the feasible region. Our key claim is that the resulting linear program is not unbounded. For edges e with $f_e \leq f_e^*$, the claim is obvious from the monotonicity constraints. For edges e with $f_e > f_e^*$, the claim follows from

Lemma 6.1 and the fact that all parameters on the right-hand side of the bound (7) are fixed in the linear program.

Since the maximum of the above linear program is bounded, we can apply Proposition 6.3. 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, $\ell_e(f_e^*) \leq M(f^*)$ for all edges e. Hence, Proposition 6.3 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 $\ell_e(f_e) \leq mn^{O(m)} \cdot M(f^*)$ for all edges e. Since a flow path of f can contain only n edges, we can conclude that $\rho(G, r, \ell) \leq nmn^{O(m)} = 2^{O(m \log n)}$. ∎

When the number of commodities is small, we can obtain a nearly optimal bound of $2^{O(kn)}$. We obtain the bound by applying Proposition 6.3 to an alternative linear program written in terms of distance variables that correspond to the lengths of shortest paths with respect to the edge lengths $\{\hat{\ell}_e(f_e)\}_{e \in E}$.

THEOREM 6.4 (upper bound #2 on the price of anarchy). *If (G, r, ℓ) is an n-vertex, k-commodity instance, then $\rho(G, r, \ell) = 2^{O(kn)}$.*

Proof. To prove our bound, we start with the linear program from the proof of Theorem 6.2. We leave the monotonicity constraints the same but change the constraints that ensure that f is a Nash equilibrium with respect to the edge latencies $\{\hat{\ell}_e(f_e)\}_{e \in E}$. The point of this change is to bring the number λ of variables in each constraint down to a constant.

We introduce an auxiliary variable $\hat{d}^i(v)$ for each commodity i and for each vertex v reachable from that commodity's source s_i, which represents the length of the shortest path from s_i to v with respect to the latencies $\{\hat{\ell}_e(f_e)\}_{e \in E}$. We use the following constraints:

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

We first prove that the feasible region of this linear program faithfully encodes the edge latencies for which f is a Nash equilibrium and then show how to reduce the number of variables of the linear program. Consider edge latencies $\{\hat{\ell}_e(f_e)\}_{e \in E}$ for which f is at Nash equilibrium. We can extend these into a feasible solution to the linear program as follows: for each commodity i and vertex v reachable from s_i, take $\hat{d}_i(v)$ to be the shortest length of an s_i-v path with respect to these latencies. Conversely, consider a feasible solution to the linear program. The constraints ensure that, for every commodity i and path $P \in \mathcal{P}_i$, the latency $\sum_{e \in P} \hat{\ell}_e(f_e)$ of this path is at least $\hat{d}_i(t_i)$; and this lower bound holds with equality if $f_e^{(i)} > 0$ for every $e \in P$. By Proposition 2.5, f is at Nash equilibrium with respect to the edge latencies $\{\hat{\ell}_e(f_e)\}_{e \in E}$. Thus, maximizing the value of $\hat{\ell}_e(f_e)$ over this feasible region yields the largest edge latency possible for e subject to f being at Nash equilibrium with respect to the computed edge latencies (and subject to the monotonicity constraints).

Currently, our linear program has $O(m + kn)$ variables. To reduce this number, we show how to eliminate the latency variables. For an edge $e = (u, v)$ with $f_e^{(i)} > 0$, $\hat{d}_i(v) = \hat{d}_i(u) + \hat{\ell}_e(f_e)$ and we can replace every occurrence of $\hat{\ell}_e(f_e)$ in our linear program with $\hat{d}_i(v) - \hat{d}_i(u)$. For every other edge e (with $f_e = 0$ and $f_e^* > 0$), $\hat{\ell}_e(f_e)$ may as well be fixed to its largest feasible value, $\ell_e(f_e^*)$ (which is at most $M(f^*)$). With these substitutions, we have not changed the optimal value of our linear program, and we are left with only $O(kn)$ variables. Moreover, there is still a constant number of variables per constraint, and the magnitude of each entry of b is still bounded by

$M(f^*)$. Proceeding as in the proof of Theorem 6.2, we can apply Proposition 6.3 and upper bound the price of anarchy of the original instance (G, r, ℓ) by $2^{O(kn)}$. ◻

7. Exponential inapproximability for network design. In this section, we show that a natural network design problem motivated by Braess's paradox has intrinsically exponential approximability. The problem, which we call Multicommodity Network Design (MCND), is as follows:

Given an instance (G, r, ℓ), find a subgraph H of G that minimizes $\max_i L_i(H, r, \ell)$.

For single-commodity instances, the best-possible polynomial-time approximation ratio (assuming $P \neq NP$) was shown to be $\lfloor n/2 \rfloor$ in [29], where n denotes the number of network vertices.

The *trivial algorithm* is defined as the algorithm that always returns the entire graph G. Theorems 6.2 and 6.4 imply that the trivial algorithm is a $2^{O(\min\{kn, m \log n\})}$-approximation algorithm for MCND. Our main result in this section uses our lower bound construction in section 5 to show that the MCND problem is approximation-resistant, in the sense that no significantly better polynomial-time approximation algorithm exists (assuming $P \neq NP$).

Theorem 7.1 (exponential inapproximability for detecting Braess's paradox). *Assuming $P \neq NP$, there is no $2^{o(n)}$-approximation algorithm for* MCND.

Proof. The idea of the reduction is to start with an instance (G^p, r^p, ℓ^p) of the form described in Theorem 5.1, and to replace the extra edge (s_1, w_0) with a collection of parallel edges representing an instance $\mathcal{I} = \{a_1, \ldots, a_q\}$ of the NP-hard problem Partition, where the feasible solutions are defined as the subsets $S \subseteq \{1, 2, \ldots, q\}$ for which $\sum_{i \in S} a_i = \frac{1}{2} \sum_{i=1}^{q} a_i$. We give these parallel edges latency functions that simulate "capacities," with an edge representing an integer a_j of $\mathcal{I}$ receiving capacity a_j. The proof has three parts. First, if too many of these parallel edges are removed from the network, there is insufficient capacity remaining to send flow cheaply. To implement this, we also give capacities to the other edges of (G^p, r^p, ℓ^p). 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 4. 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 flow at Nash equilibrium similar to that of Figure 3.

Formally, consider an instance $\mathcal{I} = \{a_j\}_{j=1}^{q}$ of Partition, with each a_j a positive integer. By scaling, there is no loss of generality in assuming that each a_j is a multiple of a large number R (say $R = q^2$). Let G denote the graph G^p from the construction in Theorem 5.1, where $p \in \{q, q+1\}$ is odd, except with the extra edge (s_1, w_0) replaced by q edges $e^1, e^2, \ldots, e^q$. We call these the *parallel edges* of G. Label the s_1-t_1 short paths $P_1, Q_0^1, \ldots, Q_0^q, Q_2, Q_4, \ldots, Q_{p-1}$ and the s_2-t_2 short paths $P_2, Q_1, Q_3, \ldots, Q_{p-2}$.

A *good flow* in G has the following form: a total of $A/2$ units of flow are routed on the paths $Q_0^1, \ldots, Q_0^q$, and R units of flow are routed on every other short path (where $R = q^2$). We set the traffic rates r_1, r_2 so that a good flow is feasible for them (so $r_1 = \frac{A}{2} + R(p+1)/2$ and $r_2 = R(p+1)/2$). Note that every good flow routes the same amount of traffic on a nonparallel edge e; denote this amount by g_e.

Also, for a nonparallel edge e, let h_e denote the difference between the amount of flow routed on e in the proof of Lemma 5.6 (Figure 4) and in the proof of Theorem 5.1(d) (Figure 3). Recall the following properties: $h_e \leq 1$ for every nonparallel edge e; $h_e > 0$ for every edge of the form e_i (as shown in (6)); and $h_e < 0$ for every nonparallel edge not of the form e_i.

We define the network latency functions as follows, for a sufficiently small constant δ:

- The parallel edge e^j is given a latency function ℓ with $\ell(x) = 0$ for $x \leq a_j - \delta$; $\ell(a_j) = 1$; and $\ell(x) = F_p$ for $x \geq a_j + \delta$.
- An edge of the form e_i is given a latency function ℓ with $\ell(x) = 0$ for $x \leq g_e + h_e - \delta$; $\ell(g_e + h_e) = F_i$; and $\ell(x) = F_p$ for $x \geq g_e + h_e + \delta$.
- Every other edge is given a latency function ℓ with $\ell(x) = 0$ for $x \leq g_e$; and $\ell(x) = F_p$ for $x \geq g_e + \delta$.

These latency functions can be defined arbitrarily outside the prescribed regions, subject to continuity and monotonicity. The *capacity* of an edge of the first type is a_j; the other edges' capacities are defined analogously. We call an edge *oversaturated* by a flow if the amount of flow on it exceeds its capacity by δ or more, in which case the maximum latency of the flow is at least F_p. The instance (G, r, ℓ) can be constructed in time polynomial in the size of the PARTITION instance $\mathcal{I}$.

The following two statements imply the theorem:

(i) If $\mathcal{I}$ is a "yes" instance of PARTITION, then G admits a subgraph H with $L_1(H, r, \ell) = 1$ and $L_2(H, r, \ell) = 0$.

(ii) If $\mathcal{I}$ is a "no" instance, then $\max_{i=1,2} L_i(H, r, \ell) \geq F_p$ for every subgraph H of G.

To prove (i), suppose that $\mathcal{I}$ admits a partition, with $\sum_{j \in S} a_j = A/2$ for some $S \subseteq \{1, 2, \ldots, q\}$. Obtain H from G by deleting all parallel edges e^j with $j \notin S$. Let f denote the corresponding good flow in H, with a_j units of flow routed on path Q_0^j for each $j \in S$. The latencies of all edges are the same as in Figure 3, with all remaining parallel edges (e^j with $j \in S$) having latency 1. Thus f is a flow at Nash equilibrium for (H, r, ℓ), which shows that $L_1(H, r, \ell) = 1$ and $L_2(H, r, \ell) = 0$.

For statement (ii), we first claim that if H omits any nonparallel edge, then every feasible flow in H oversaturates some edge and hence has maximum latency at least F_p. The basic reason for this is the following: edge capacities ensure that every flow which oversaturates no edge routes at most $\max\{0, h_e\} + \delta \leq 1 + \delta$ more units of flow than a good flow on every nonparallel edge; and removing a nonparallel edge destroys some short path, thereby forcing at least R units of flow to be rerouted relative to a good flow. For example, suppose H omits some edge incident to s_2. This reduces the total edge capacity incident to s_2 to at most $\frac{1}{2}(p-1)(R+1)$, whereas the traffic rate r_2 is $\frac{1}{2}(p+1)R$. Assuming R is sufficiently large and δ is sufficiently small, every feasible flow then oversaturates an edge incident to s_2. Analogous arguments show that every nonparallel edge removal inevitably leads to an oversaturated edge.

Now suppose that H omits only parallel edges. Since $\mathcal{I}$ is a "no" instance and all a_j's are multiples of R, the total capacity of the remaining edges incident to s_1 is either strictly more or strictly less than $r_1 = \frac{A}{2} + R(p+1)/2$ by at least R. In the latter case, a capacity argument again shows that every feasible flow oversaturates some edge incident to s_1.

For the final case, in which the total capacity of the parallel edges in H is $A' \geq \frac{A}{2} + R$, we explicitly define a flow f at Nash equilibrium for (H, r, ℓ) with the same edge latencies as in Figure 4, with all remaining parallel edges having zero latency with respect to f. Intuitively, we define f as follows: we begin with a good flow g, with the flow on the parallel edges split in proportion to their capacities; then, we take one unit of flow off of P_1 and P_2 and replace it with the (bad) flow at Nash equilibrium shown in Figure 4, again splitting the additional flow on the parallel edges in proportion to the edge capacities. Precisely, we set $f_e = g_e + h_e$ for every nonparallel edge e; and for every parallel edge e^j of H, we set

$$f_{e_j} = \frac{a_j}{A'} \cdot \left(\frac{A}{2} + h_{sw_0}\right),$$

where h_{sw_0} is the flow on the extra edge (s, w_0) in the proof of Lemma 5.6. Since $A' \geq \frac{A}{2} + R$, $f_{e_j} \leq a_j - \delta$ for large enough R and small enough δ, so the latency of edge e_j is 0. Thus, f is a feasible flow for (H, r, ℓ) with edge latencies precisely as in Figure 4; it is therefore at Nash equilibrium and proves that $L_2(H, r, \ell) \geq F_p$. □

Acknowledgment. We thank Hisao Kameda for sharing his conjecture (Corollary 4.2) with us.

REFERENCES

[1] Y. Azar and A. Epstein, *The hardness of network design for unsplittable flow with selfish users*, in Proceedings of the Third Workshop on Approximation and Online Algorithms (WAOA), 2005, pp. 41–54.

[2] R. Banner and A. Orda, *Bottleneck routing games in communication networks*, IEEE J. Sel. Area Commun., 25 (2007), pp. 1173–1179.

[3] M. J. Beckmann, C. B. McGuire, and C. B. Winsten, *Studies in the Economics of Transportation*, Yale University Press, New Haven, CT, 1956.

[4] D. Braess, *Über ein Paradoxon aus der Verkehrsplanung*, Unternehmensforschung, 12 (1968), pp. 258–268.

[5] C. Busch and M. Magdon-Ismail, *Atomic routing games on maximum congestion*, Theoret. Comput. Sci., 410 (2009), pp. 3337–3347.

[6] D. Chakrabarty, A. Mehta, V. Nagarajan, and V. Vazirani, *Fairness and optimality in congestion games*, in Proceedings of the 6th ACM Conference on Electronic Commerce (EC), 2005, pp. 67–73.

[7] C. K. Chau and K. M. Sim, *The price of anarchy for non-atomic congestion games with symmetric cost maps and elastic demands*, Oper. Res. Lett., 31 (2003), pp. 327–334.

[8] G. Christodoulou and E. Koutsoupias, *The price of anarchy of finite congestion games*, in Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC), 2005, pp. 67–73.

[9] V. Chvátal, *Linear Programming*, Freeman, New York, 1983.

[10] J. R. Correa, A. S. Schulz, and N. E. Stier Moses, *Selfish routing in capacitated networks*, Math. Oper. Res., 29 (2004), pp. 961–976.

[11] J. R. Correa, A. S. Schulz, and N. E. Stier Moses, *Fast, fair, and efficient flows in networks*, Oper. Res., 55 (2007), pp. 215–225.

[12] J. R. Correa, A. S. Schulz, and N. E. Stier Moses, *On the inefficiency of equilibria in congestion games*, Games Econom. Behav., 64 (2008), pp. 457–469.

[13] D. Dumrauf and M. Gairing, *Price of anarchy for polynomial Wardrop games*, in Proceedings of the 2nd Annual Workshop on Internet and Network Economics (WINE), Lecture Notes in Comput. Sci. 4286, Springer, Berlin, 2006, pp. 319–330.

[14] A. Epstein, M. Feldman, and Y. Mansour, *Efficient graph topologies in network routing games*, Games Econom. Behav., 66 (2009), pp. 115–125.

[15] M. A. Hall, *Properties of the equilibrium state in transportation networks*, Trans. Sci., 12 (1978), pp. 208–216.

[16] O. Jahn, R. H. Möhring, A. S. Schulz, and N. E. Stier Moses, *System-optimal routing of traffic flows with user constraints in networks with congestion*, Oper. Res., 53 (2005), pp. 600–616.

[17] H. Kameda, *private communication*, 2002.

[18] H. Kameda, *How harmful the paradox can be in the Braess/Cohen-Kelly-Jeffries networks*, in Proceedings of the 21st INFOCOM Conference, Vol. 1, 2002, pp. 437–445.

[19] E. Köhler and M. Skutella, *Flows over time with load-dependent transit times*, SIAM J. Optim., 15 (2005), pp. 1185–1202.

[20] E. Koutsoupias and C. H. Papadimitriou, *Worst-case equilibria*, in Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS), Lecture Notes in Comput. Sci. 1563, Springer, Berlin, 1999, pp. 404–413.

[21] 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), 2004, pp. 333–334.
[22] H. Lin, T. Roughgarden, É. Tardos, and A. Walkover, *Braess's paradox, Fibonacci numbers, and exponential inapproximability*, in Proceedings of the 32nd Annual International Colloquium on Automata, Languages, and Programming (ICALP), Lecture Notes in Comput. Sci. 3580, Springer, Berlin, 2005, pp. 497–512.
[23] N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, eds., *Algorithmic Game Theory*, Cambridge University Press, Cambridge, UK, 2007.
[24] C. H. Papadimitriou, *Algorithms, games, and the internet*, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC), 2001, pp. 749–753.
[25] G. Perakis, *The "price of anarchy" under nonlinear and asymmetric costs*, Math. Oper. Res., 32 (2007), pp. 614–628.
[26] T. Roughgarden, *How unfair is optimal routing?*, in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2002, pp. 203–204.
[27] T. Roughgarden, *The price of anarchy is independent of the network topology*, J. Comput. System Sci., 67 (2003), pp. 341–364.
[28] T. Roughgarden, *The maximum latency of selfish routing*, in Proceedings of the 15th Annual Symposium on Discrete Algorithms (SODA), 2004, pp. 973–974.
[29] T. Roughgarden, *On the severity of Braess's Paradox: Designing networks for selfish users is hard*, J. Comput. System Sci., 72 (2006), pp. 922–953.
[30] T. Roughgarden and É. Tardos, *How bad is selfish routing?*, J. ACM, 49 (2002), pp. 236–259.
[31] M. J. Smith, *The existence, uniqueness and stability of traffic equilibria*, Transportation Res. Part B, 13 (1979), pp. 295–304.
[32] J. G. Wardrop, *Some theoretical aspects of road traffic research*, in Proceedings of the Institute of Civil Engineers, Pt. II, Vol. 1, 1952, pp. 325–378.
[33] D. Weitz, *The Price of Anarchy*, manuscript, 2001.