Routing Without Regret: On Convergence to Nash Equilibria of Regret-Minimizing Algorithms in Routing Games

Avrim Blum [1] Eyal Even-Dar [2] Katrina Ligett [3]

February 2006
CMU-CS-06-107

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

There has been substantial work developing simple, efficient *no-regret* algorithms for a wide class of repeated decision-making problems including online routing. These are adaptive strategies an individual can use that give strong guarantees on performance even in adversarially-changing environments. There has also been substantial work on analyzing properties of Nash equilibria in routing games. In this paper, we consider the question: if each player in a routing game uses a no-regret strategy, will behavior converge to a Nash equilibrium? In general games the answer to this question is known to be *no* in a strong sense, but routing games have substantially more structure.

In this paper we show that in the Wardrop setting of multicommodity flow and infinitesimal agents, behavior will approach Nash equilibrium (formally, on most days, the cost of the flow will be close to the cost of the cheapest paths possible given that flow) at a rate that depends polynomially on the players' regret bounds and the maximum slope of any latency function. We also show that price-of-anarchy results may be applied to these approximate equilibria, and also consider the finite-size (non-infinitesimal) load-balancing model of Azar [2].

[1] Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. `avrim@cs.cmu.edu`. Tel: (412) 268-6452. Fax: (412) 268-5576. Supported in part by the National Science Foundation under grants IIS-0121678 and CCR-0122581.

[2] School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel 69978 and University of Pennsylvania, Department of Information and Computer Science, Philadelphia, PA 19104. `evendar@seas.upenn.edu`.

[3] Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. `katrina@cs.cmu.edu`. Supported in part by an AT&T Labs Graduate Fellowship.

Keywords: game theory, network games

1 Introduction

There has been substantial work in learning theory and game theory on adaptive *no-regret* algorithms for problems of repeated decision-making. These algorithms have the property that in any online, repeated game setting, their average loss per time step approaches that of the best fixed strategy in hindsight (or better) over time. Moreover, the convergence rates are quite good: in Hannan's original algorithm [18], the number of time steps needed to achieve a gap of ϵ with respect to the best fixed strategy in hindsight—the "per time step regret"—is linear in the size of the game N. This was reduced to $O(\log N)$ in more recent exponential-weighting algorithms for this problem [22, 5, 15] (also called the problem of "combining expert advice"). Most recently, a number of algorithms have been developed for achieving such guarantees *efficiently* in many settings where the number of choices N is exponential in the natural description-length of the problem [20, 27, 29].

One specific setting where these efficient algorithms apply is online routing. Given a graph $G = (V, E)$ and two distinguished nodes v_{start} and v_{end}, the game for an individual player is defined as follows. At each time step t, the player's algorithm chooses a path P_t from v_{start} to v_{end}, and simultaneously an adversary (or nature) chooses a set of edge costs $\{c_e^t\}_{e \in E}$. The edge costs are then revealed and the player pays the cost of its path. Even though the number of possible paths can be exponential in the size of the graph, no-regret algorithms exist (e.g., [20, 29]) that achieve running time and convergence rates (to the cost of the best fixed path in hindsight) which are polynomial in the size of the graph and the maximum edge cost. Moreover, a number of extensions [1, 23] have shown how these algorithms can be applied even to the "bandit" setting where only the cost of edges actually traversed (or even just the total cost of P_t) is revealed to the algorithm.

Along a very different line of inquiry, there has also been much recent work on *price of anarchy* results. Koutsoupias and Papadimitriou [21] defined the *price of anarchy*, which is the ratio of the cost of an optimal global objective function to the cost of the worst Nash equilibrium. Many subsequent results have studied the price of anarchy in a wide range of computational problems from job scheduling to facility location to network creation games, and especially to problems of routing in the Wardrop model, where the cost of an edge is a function of the amount of traffic using that edge [6, 7, 21, 25, 10]. Such work implicitly assumes that selfish individual behavior results in Nash equilibria.

In this work we consider the question: if all players in a routing game use no-regret algorithms to choose their paths each day, what can we say about the overall behavior of the system? In particular, the no-regret property (also called Hannan Consistency) can be viewed as a natural *definition* of well-reasoned self-interested behavior over time. Thus, if all players are adapting their behavior in such a way, can we say that the system as a whole will approach Nash equilibrium? Our main result is that in the Wardrop setting of multicommodity flow and infinitesimal agents, the flows will approach equilibrium in the sense that a $1 - \epsilon$ fraction of the daily flows will have the property that at most an ϵ fraction of the agents in them have more than an ϵ incentive to deviate from their chosen path, where ϵ approaches 0 at a rate that depends polynomially on the size of the graph, the regret-bounds of the algorithms, and the maximum slope of any latency function.[1] Moreover, we show that the one new parameter—the dependence on slope—is necessary. In addition, we give stronger results for special cases such as the case of n parallel links and also consider the finite-size (non-infinitesimal) load-balancing model of Azar [2].

One way our result can be viewed is as follows. No-regret algorithms are very compelling from the point of view of individuals: if you use a no-regret algorithm to drive to work each day, you will get a good

[1] A more traditional notion of approximate Nash equilibrium requires that *no* player will have more than ϵ incentive to deviate from her strategy. However, one cannot hope to achieve such a guarantee using arbitrary no-regret algorithms, since such algorithms allow players to occasionally try bad paths, and in fact such experimentation is even necessary in bandit settings. For the same reason, one cannot hope that *all* days will be approximate-Nash. Finally, our guarantee may make one worry that some users could always do badly, falling in the ϵ minority on every day, but as we discuss in Section 5, the no-regret property can be used to further show that no player experiences many days in which her expected cost is much worse than the best path available on that day.

guarantee on your performance no matter what is causing congestion (other drivers, road construction, or unpredictable events). But it would be a shame if, were everyone to use such an algorithm, this produced globally unstable behavior. Our results imply that in the Wardrop routing model, so long as edge latencies have bounded slope, we can view Nash equilibria as not just a stable steady-state or the result of adaptive procedures specifically designed to find them, but in fact as the inevitable result of individually-selfish adaptive behavior by agents that do not necessarily know (or care) what policies other agents are using. Moreover, our results do not in fact require that users follows strategies that are no-regret in the worst-case, as long as their behavior satisfies the no-regret property over the sequence of flows actually observed.

Regret and Nash equilibria: At first glance, a result of this form seems that it should be obvious given that a Nash equilibrium is precisely a set of (pure or mixed) strategies that are all no-regret with respect to each other. Thus if the learning algorithms settle at all, they will have to settle at a Nash equilibrium. In fact, for *zero-sum* games, no-regret algorithms when played against each other will approach a minimax optimal solution [16]. However, it is known that even in small 2-player *general-sum* games, no-regret algorithms need not approach a Nash equilibrium and can instead cycle, achieving performance substantially worse than any Nash equilibrium for all players. Indeed simple examples are known where standard algorithms will have this property with arbitrarily high probability [30].

Regret and Correlated equilibria: It is known that certain algorithms such as that of Hart and Mas-Colell [19], as well as any algorithms satisfying the stronger property of “no internal regret” [14], have the property that the empirical distribution of play approaches a *correlated* equilibrium. On the positive side, such results are extremely general, apply to nearly any game including routing, and do not require any bound on the slopes of edge latencies. However, such results do *not* imply that the daily flows themselves (or even the average flow) are at all close to equilibrium. It could well be that on each day, a substantial fraction of the players experience latency substantially greater than the best path given the flow (and we give a specific example of how this can happen when edge-latencies have unbounded slope in Section 2.4).

Related work: Fischer and Vöcking [12] consider a specific adaptive dynamics (a particular functional form in which flow might naturally change over time) in the context of selfish routing and prove results about convergence of this dynamics to an approximately stable configuration. In more recent work, they study the convergence of a class of routing policies under a specific model of stale information [13]. Most recently, Fischer, Raecke, and Vöcking [11] give a distributed procedure with especially good convergence properties. The key difference between that work and ours is that those results consider specific adaptive strategies designed to quickly approach equilibrium. In contrast, we are interested in showing convergence for *any* algorithms satisfying the no-regret property. That is, even if each player is using a different strategy, without necessarily knowing or caring about what strategies others are using, then so long as all are no-regret, we show they achieve convergence. In addition, because efficient no-regret algorithms exist even in the bandit setting where each agent gets feedback only about its own actions [1, 23], our results can apply to scenarios in which agents adapt their behavior based on only very limited information and there is no communication at all between different agents.

Convergence time to Nash equilibrium in load balancing has also been studied. Earlier work studied convergence time using potential functions, with the limitation that only one player is allowed to move in each time step; the convergence times derived depended on the appropriate potential functions of the exact model [24, 8]. The work of Goldberg [17] studied a randomized model in which each user can select a random delay over continuous time. This implies that only one user tries to reroute at each specific time; therefore the setting was similar to that mentioned above. Even-Dar and Mansour [9] considered a model where many users are allowed to move concurrently, and derived a logarithmic convergence rate for users following a centrally-moderated greedy algorithm. Most recently, Berenbrink et al. [4] showed weaker convergence results for a specific distributed protocol. To summarize, previous work studied the convergence time to pure Nash equilibria in situations with a centralized mechanism or specific protocol. In contrast, we present fast convergence results for approximate Nash equilibria in a non-centralized setting, and our only assumption about the player strategies is that they are all no-regret.

Structure of this paper: For ease of exposition we first discuss the special case of single-commodity flow, where all users share the same start node and end node. We begin by focusing on the time-average flow, analyzing how that approaches equilibrium, and then use those results to prove convergence of the flows at each time step. In Section 7 we show how to extend our results to the general case of multicommodity flow, where different users may have different start and end nodes, and even different subsets of allowable edges. This model generalizes both multicommodity flow and the parallel links restricted-machines model, and can also model the notion that users traveling at different times of day may not affect each other.

2 Preliminaries

2.1 The Model

When dealing with networks and flows, we adopt the notation used by Roughgarden [26], which we summarize here. The definitions in this section pertain to single-commodity flow; necessary changes for the multicommodity flow setting are discussed in Section 7.

Let $G = (V, E)$ be a directed network with a source vertex v_{start} and a sink vertex v_{end}. We allow multi-edges but disallow self-loops, as self-loops are redundant in this context. Let n be the number of nodes in the network, and let m be the number of edges. Let $\mathcal{P}$ represent the set of simple v_{start}-v_{end} paths on G. A *flow* is a function $f : \mathcal{P} \to \mathcal{R}^+$, such that $\sum_{P \in \mathcal{P}} f_P = 1$ (instances with other traffic rates may be normalized accordingly). Each flow induces a unique flow on edges such that the flow f_e on an edge e has the property $f_e = \sum_{P: e \in P} f_P$. Each edge $e \in E$ has an associated traffic-dependent, positive, continuous, non-decreasing *latency* function ℓ_e. The latency of a path P given a flow f is $\ell_P(f) = \sum_{e \in P} \ell_e(f_e)$, i.e., the sum of the latencies of the edges in the path, given that flow.

Let $f^1, f^2, \ldots, f^T$ denote a series of flows from time 1 up to time T. We use $\hat{f}$ to denote the time-average flow, i.e., $\hat{f}_e = \frac{1}{T} \sum_{t=1}^{T} f_e^t$.

We will assume all edge latency functions have range $[0, 1]$, so the latency of a path in $\mathcal{P}$ is always between 0 and $n - 1$.

2.2 Flows at Nash Equilibria

A flow f is at *Nash equilibrium* if no user would prefer to reroute her traffic, given the existing flow.

Proposition 2.1. *(Wardrop [28]) A flow f is at Nash equilibrium if and only if for every $P_1, P_2 \in \mathcal{P}$ with $f_{P_1} > 0$, $\ell_{P_1}(f) \leq \ell_{P_2}(f)$.*

It is useful to note that in this domain, the flows at Nash equilibrium are those for which all flow-carrying paths have the same latency. In addition, given our assumption that all latency functions are continuous and non-decreasing, one can prove the existence and uniqueness of Nash equilibria:

Proposition 2.2. *(Beckman et al. [3]) For every directed graph G, there exists a Nash flow. Moreover, if f, f' are Nash flows then $\ell_P(f) = \ell_P(f')$ for every v_{start}-v_{end} path P.*

We define the cost of a flow to be the average latency incurred by users on that flow:

Definition 2.3. Define the *cost* $C(f)$ of a flow f to be $C(f) = \sum_{e \in E} \ell_e(f_e) f_e$.

2.3 No-Regret Algorithms

Definition 2.4. Consider a series of flows $f^1, f^2, \ldots, f^T$ and a user who has experienced latencies $c^1, c^2, \ldots, c^T$ over these flows. The per-time-step *regret* of the user is the difference between her average latency and the latency of the best fixed path in hindsight, that is,

$$\frac{1}{T}\sum_{t=1}^{T} c^t - \min_{P \in \mathcal{P}} \frac{1}{T}\sum_{t=1}^{T}\sum_{e \in P} \ell_e(f_e^t).$$

An algorithm is *no-regret* if, for any sequence of flows, the expected regret (over internal randomness in the algorithm) goes to 0 as T goes to infinity.

Here and in the rest of this paper, excluding Section 8, we consider infinitesimal users using a finite number of different algorithms; in this setting, we can get rid of the expectation. In particular, if each user is running a no-regret algorithm, then the average regret over users also approaches 0. Thus, this means we can make the following assumption:

Assumption 2.5. The series of flows $f^1, f^2, \ldots$ satisfies

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{e \in E} \ell_e(f_e^t) f_e^t \leq R(T) + \frac{1}{T}\min_{P \in \mathcal{P}}\sum_{t=1}^{T}\sum_{e \in P} \ell_e(f_e^t)$$

where $R(T) \to 0$ as $T \to \infty$. The function $R(T)$ may depend on the size of the network and its maximum possible latency. We then define T_ϵ as the number of time steps required to get $R(T) = \epsilon$.

For example, for the case of a network consisting of only two nodes and m parallel edges, exponential-weighting algorithms [22, 5, 15] give $T_\epsilon = O(\frac{1}{\epsilon^2}\log m)$. For general graphs, results of Kalai and Vempala yield $T_\epsilon = O(\frac{mn \log n}{\epsilon^2})$ [20]. For general graphs where an agent can observe only its path cost, results of Awerbuch and Kleinberg yield $T_\epsilon = \tilde{O}(\frac{d^7 m}{\epsilon^3})$, where d is the length of the longest path [1].

2.4 Approaching Nash Equilibria

We now need to specify in what sense flow will be approaching a Nash equilibrium. The first notion one might consider is the L_1 distance to some true Nash flow. However, if some edges have nearly-flat latency functions, it is possible for a flow to have regret near 0 and yet still be far in L_1 distance to a true Nash flow. A second natural notion would be to say that the flow f has the property that no user has cost much more than the cheapest path given f. However, notice that the no-regret property allows users to occasionally take long paths, so long as they perform well on average (and in fact algorithms for the bandit problem will have exploration steps that do just that [1, 23]). So, one cannot expect that on any time step *all* users are taking cheap paths.

Instead, we require that *most* users be taking a nearly-cheapest path given f. Specifically,

Definition 2.6. A flow f is at ϵ-Nash equilibrium if the average cost under this flow is within ϵ of the minimum cost path under this flow, i.e. $C(f) - \min_{P \in \mathcal{P}} \sum_{e \in P} \ell_e(f_e) \leq \epsilon$.

Note that Definition 2.6 implies that at most a $\sqrt{\epsilon}$ fraction of traffic can have more than a $\sqrt{\epsilon}$ incentive to deviate from their path, and as a result is very similar to the definition of (ϵ, δ)-Nash equilibria in [11].

We also are able to show that one can apply price-of-anarchy results to ϵ-Nash flows; we discuss this in Section 6.

We will begin by focusing on the *time-average* flow $\hat{f}$, showing that for no-regret algorithms, this flow is approaching equilibrium. That is, for a given T_ϵ we will give bounds on the number of time steps before $\hat{f}$ is ϵ-Nash. After analyzing $\hat{f}$, we then extend our analysis to show that in fact for *most* time steps t, the flow f^t itself is ϵ-Nash. To achieve bounds of this form, which we show in Section 5, we will however need to lose an additional factor polynomial in the size of the graph. Again, we cannot hope to say that f^t is ϵ-Nash for *all* (sufficiently large) time-steps t, because no-regret algorithms may occasionally take long paths, and an "adversarial" set of such algorithms may occasionally all take long paths at the same time.

Dependence on slope: Our convergence rates will depend on the maximum slope s allowed for any latency function. To see why this is necessary, consider the case of two parallel links, where one edge has latency 0 up to a load of $1/3$ and then rises immediately to 1, and the other edge has latency 0 up to a load of $2/3$ and then rises directly to 1. In this case the Nash cost is 0, and moreover for *any* flow f' we have $\min_{P \in \mathcal{P}} \sum_{e \in P} \ell_e(f'_e) = 0$. Thus, the only way f' can be ϵ-Nash is for it to actually have low cost, which means the algorithm must precisely be at a $1/3$-$2/3$ split. If players use no-regret algorithms, traffic will instead oscillate, each edge having cost 1 on about half the days and each player incurring cost 1 on not much more than half the days (and thus not having much regret). However, none of the daily flows will be ϵ-Nash.

3 Infinitesimal Users: Linear Latency Functions

We begin as a warm-up with the easiest case, infinitesimal users and linear latency functions, which simplifies many of the arguments. In particular, for linear latency functions, the latency of any edge given the average flow $\hat{f}$ is guaranteed to be equal to the average latency of that edge over time, i.e. $\ell_e(\hat{f}_e) = \frac{1}{T}\sum_{t=1}^{T} \ell_e(f_e^t)$ for all e.

Theorem 3.1. *Suppose the latency functions are linear. Then for $T \geq T_\epsilon$, the average flow $\hat{f}$ is ϵ-Nash, i.e.*

$$C(\hat{f}) \leq \epsilon + \min_P \sum_{e \in P} \ell_e(\hat{f}_e).$$

Proof. From the linearity of the latency functions, we have for all e, $\ell_e(\hat{f}_e) = \frac{1}{T}\sum_{t=1}^{T} \ell_e(f_e^t)$. Since $\ell_e(f_e^t) f_e^t$ is a convex function of the flow, this implies $\ell_e(\hat{f}_e)\hat{f}_e \leq \frac{1}{T}\sum_{t=1}^{T} \ell_e(f_e^t) f_e^t$. Summing over all e, we have

$$\begin{array}{rcll} C(\hat{f}) & \leq & \frac{1}{T}\sum_{t=1}^{T} C(f^t) & \\ & \leq & \epsilon + \min_P \frac{1}{T}\sum_{t=1}^{T}\sum_{e\in P} \ell_e(f_e^t) & \text{(by Assumption 2.5)} \\ & = & \epsilon + \min_P \sum_{e\in P} \ell_e(\hat{f}_e). & \text{(by linearity)} \end{array}$$

□

Corollary 3.2. *Assume that all latency functions are linear. On general graphs, if all agents use the Kalai-Vempala algorithm [20], the average flow converges to an ϵ-Nash equilibrium at $T_\epsilon = O(\frac{mn \log n}{\epsilon^2})$. On networks consisting of two nodes and m parallel links, if all agents use optimized "combining expert advice"-style algorithms (with each edge an expert), the average flow converges to an ϵ-Nash equilibrium at $T_\epsilon = O(\frac{\log m}{\epsilon^2})$.*

Note that we not only proved that the average flow approaches an ϵ-Nash equilibrium, but as an intermediate step in our proof we showed that *actual* average cost incurred by the users is at most ϵ worse than the best path in the average flow.

4 Infinitesimal Users: General Latency Functions

The case of general latency functions is more complicated because the first and third transitions in the proof above do not apply. Here, the additive term depends on the maximum slope of any latency function.

Theorem 4.1. *Let $\epsilon' = \epsilon + 2\sqrt{s\epsilon n}$. Then for general functions with maximum slope s, for $T \geq T_\epsilon$, the average flow is ϵ'-Nash, i.e.*

$$\sum_{e \in E} \ell_e(\hat{f}_e)\hat{f}_e \leq \epsilon + 2\sqrt{s\epsilon n} + \min_P \sum_{e \in P} \ell_e(\hat{f}_e)$$

Before giving the proof, we list several quantities we will need to relate:

$$\text{(the cost of } \hat{f}\text{)} \qquad \sum_{e \in E} \ell_e(\hat{f}_e)\hat{f}_e \tag{1}$$

$$\text{(the ``cost of } \hat{f} \text{ in hindsight'')} \qquad \frac{1}{T}\sum_{t=1}^{T}\sum_{e \in E} \ell_e(f_e^t)\hat{f}_e \tag{2}$$

$$\text{(the average cost of flows up to time } T\text{)} \qquad \frac{1}{T}\sum_{t=1}^{T}\sum_{e \in E} \ell_e(f_e^t)f_e^t \tag{3}$$

$$\text{(the cost of the best path in hindsight)} \qquad \min_P \sum_{e \in P} \frac{1}{T}\sum_{t=1}^{T} \ell_e(f_e^t) \tag{4}$$

$$\text{(the cost of the best path given } \hat{f}\text{)} \qquad \min_P \sum_{e \in P} \ell_e(\hat{f}_e) \tag{5}$$

We now begin with a lemma:

Lemma 4.2. *For general latency functions with maximum slope s,* $(4) \leq \sqrt{s\epsilon n} + (5)$.

Proof of Lemma 4.2. First, observe that, because our latency functions are non-decreasing, the average latency of an edge must be less than or equal to the latency of that edge as seen by a random user on a random day. That is, for all e, $\hat{f}_e \sum_{t=1}^{T} \ell_e(f_e^t) \leq \sum_{t=1}^{T} \ell_e(f_e^t) f_e^t$.

If we define $\sum_{e \in E} \epsilon_e = \epsilon$, this gives us

$$\epsilon_e + \frac{1}{T}\sum_{t=1}^{T} \ell_e(f_e^t)\hat{f}_e \geq \frac{1}{T}\sum_{t=1}^{T} \ell_e(f_e^t)f_e^t \geq \frac{1}{T}\sum_{t=1}^{T} \ell_e(f_e^t)\hat{f}_e$$

for all e, since this equation is bounded from below by (4) and from above by $\epsilon + (4)$. We can rewrite this to get

$$\epsilon_e \geq \frac{1}{T}\sum_{t=1}^{T} \ell_e(f_e^t)(f_e^t - \hat{f}_e) \geq 0$$

for all e, and thus

$$\forall e, \epsilon_e \geq \frac{1}{T}\sum_{t=1}^{T}(\ell_e(f_e^t) - \ell_e(\hat{f}_e))(f_e^t - \hat{f}_e) \geq 0.$$

This is a very useful equation, as it gives tight bounds on the relationship between the difference between the latency of the average flow on an edge and the average latency on that edge.

From the bound on the maximum slope of any latency function, we know that $|f_e^t - \hat{f}_e| \geq |\ell_e(f_e^t) - \ell_e(\hat{f}_e)|/s$ and thus

$$|\ell_e(f_e^t) - \ell_e(\hat{f}_e)| \leq \sqrt{s\left(\ell_e(f_e^t) - \ell_e(\hat{f}_e)\right)\left(f_e^t - \hat{f}_e\right)} \tag{6}$$

for all e.

By properties of variance, we then get

$$\frac{1}{T}\sum_{t=1}^{T}(\ell_e(f_e^t) - \ell_e(\hat{f}_e)) \leq \sqrt{s}\frac{1}{T}\sum_{t=1}^{T}\sqrt{(\ell_e(f_e^t) - \ell_e(\hat{f}_e))(f_e^t - \hat{f}_e)}$$

Using equation (6) above, this yields

$$\frac{1}{T}\sum_{t=1}^{T}(\ell_e(f_e^t) - \ell_e(\hat{f}_e)) \leq \sqrt{s\epsilon_e}. \tag{7}$$

This gives us $(4) \leq \sum_{e\in P}\sqrt{s\epsilon_e} + (5) \leq \sqrt{s\epsilon n} + (5)$, because in the worst case, $\epsilon_e = \frac{\epsilon}{n}$. $\square$

We now use a second lemma:

Lemma 4.3. *For general latency functions with maximum slope s, $(1) \leq \sqrt{s\epsilon n} + (2)$.*

Proof of Lemma 4.3. Equation (7) above directly gives us $(1) \leq \sum_{e\in P}\sqrt{s\epsilon_e}\hat{f}_e + (2)$. We then use the fact that $\hat{f}_e \leq 1$ for all e to obtain the desired result. $\square$

Given the above lemmas we now present the proof of Theorem 4.1.

Proof of Theorem 4.1. Since $(3) \leq \epsilon + (4)$ by Assumption 2.5, and $(2) \leq (3)$ by convexity, we get

$$(1) \leq \sqrt{s\epsilon n} + (2) \leq \sqrt{s\epsilon n} + (3) \leq \epsilon + \sqrt{s\epsilon n} + (4) \leq \epsilon + 2\sqrt{s\epsilon n} + (5)$$

as desired. $\square$

Corollary 4.4. *Let $\epsilon' = \epsilon + 2\sqrt{s\epsilon n}$. Assume that all latency functions are positive, non-decreasing, and continuous, with maximum slope s. On general graphs, if all agents use the Kalai-Vempala algorithm [20], the average flow converges to an ϵ'-Nash equilibrium at $T_\epsilon = O(\frac{mn\log n}{\epsilon^2}) = O(\frac{mn^3s^2\log n}{\epsilon'^4})$. On networks consisting of two nodes and m parallel links, if all agents use optimized "combining expert advice"-style algorithms, the average flow converges to an ϵ'-Nash equilibrium at $T_\epsilon = O(\frac{\log m}{\epsilon^2}) = O(\frac{n^2s^2\log m}{\epsilon'^4})$.*

Once again we remark that not only have we proved that the average flow approaches ϵ'-Nash equilibrium, but as an intermediate step in our proof we showed that *actual* average cost obtained by the users is at most ϵ' worse than the best path in the average flow.

5 Infinitesimal Users: How Bad is the Traffic Today?

Here we present results applicable to general graphs and general functions showing that on *most* time steps, the flow will be at ϵ-Nash equilibrium.

Theorem 5.1. *On general graphs with general latency functions with maximum slope s, for all but a $(ms^{1/4}\epsilon^{1/4})$ fraction of time steps up to time T_ϵ, f^t is a $(\epsilon + 2\sqrt{s\epsilon n} + 2m^{3/4}s^{1/4}\epsilon^{1/4})$-Nash flow. We can rewrite this as: for all but an ϵ' fraction of time steps up to T_ϵ, f^t is an ϵ'-Nash flow for $\epsilon = \Omega\left(\frac{\epsilon'^4}{sm^4+s^2n^2}\right)$.*

Proof. As shown in equation (6),

$$\sqrt{s\epsilon_e} \geq |\ell_e(f_e^t) - \ell_e(\hat{f}_e)|$$

for all edges. Thus, for all edges, for all but $s^{1/4}\epsilon_e^{1/4}$ of the time steps,

$$s^{1/4}\epsilon_e^{1/4} \geq |\ell_e(f_e^t) - \ell_e(\hat{f}_e)|.$$

Using a union bound over edges, this implies that on all but a $ms^{1/4}\epsilon^{1/4}$ fraction of the time steps, *all* edges have

$$s^{1/4}\epsilon_e^{1/4} \geq |\ell_e(f_e^t) - \ell_e(\hat{f}_e)|.$$

From this, it follows directly that on most time steps, the cost of the best path given f^t differs from the cost of the best path given $\hat{f}$ by at most $m^{3/4}s^{1/4}\epsilon^{1/4}$. Also on most time steps, the cost incurred by flow f^t differs from the cost incurred by flow $\hat{f}$ by at most $m^{3/4}s^{1/4}\epsilon^{1/4}$. Thus since $\hat{f}$ is an $(\epsilon + 2\sqrt{s\epsilon n})$-Nash equilibrium, f^t is an $(\epsilon + 2\sqrt{s\epsilon n} + 2m^{3/4}s^{1/4}\epsilon^{1/4})$-Nash equilibrium on all but a $ms^{1/4}\epsilon^{1/4}$ fraction of time steps.

□

Corollary 5.2. *On general graphs with general latency functions with maximum slope s, for all but a $(ms^{1/4}\epsilon^{1/4})$ fraction of time steps up to time $T = T_\epsilon$, the expected average cost $\frac{1}{T}\sum_{t=1}^{T} c^t$ incurred by any user is at most $(\epsilon + 2\sqrt{s\epsilon n} + m^{3/4}s^{1/4}\epsilon^{1/4})$ worse than the cost of the best path on that time step.*

This demonstrates that no-regret algorithms are incentive-compatible in a network setting: if a player knows that all other players are using no-regret algorithms, there is no strategy that will significantly improve her expected cost on more than a small fraction of days. By using a no-regret algorithm, she gets the guarantee that on most time steps her expected cost is within some epsilon of the cost of the best path given the flow for that day.

Proof sketch for Corollary 5.2: From the proof of Theorem 5.1 we see that on most days, the cost of the best path given the flow for that day is within $m^{3/4}s^{1/4}\epsilon^{1/4}$ of the cost of the best path given $\hat{f}$, which is at most $2\sqrt{s\epsilon n}$ worse than the cost of the best path in hindsight. Combining this with the no-regret property achieved by each user gives the desired result.

6 ϵ-Nash and the Price of Anarchy

In this section, we sketch how one can apply price-of-anarchy results, which bound the relationship between a Nash flow and the optimum flow, to ϵ-Nash equilibria.

Claim 6.1. *For every network G and flow f at ϵ-Nash equilibrium on G, there exists a network G' that approximates G and a flow f' that approximates f such that: (a) f' is a Nash flow on G', (b) the cost of f' on G' is at most ϵ less than the cost of f on G, and (c) the cost of the optimal flow on G' is within $\sqrt{\epsilon}$ of the cost of the optimal flow on G. These approximations allow one to apply price-of-anarchy results from f' and G' to f and G.*

Proof sketch: Note that since f is at ϵ-Nash equilibrium on G, then at most a $\sqrt{\epsilon}$ fraction of users are experiencing costs more than $\sqrt{\epsilon}$ worse than the cost of the best path given f. We can modify G to embed the costs associated with these "meandering" users such that the costs experienced by the remaining users do not change. We then rescale the latency functions and remaining flow so that we once more have one unit of flow; the total cost incurred by the rescaled flow f' on the new network has decreased by at most ϵ, since meandering users were responsible for no more than ϵ of the original cost.

Notice now that the cost c of the worst flow-carrying path is at most $\sqrt{\epsilon}$ worse than the cost of the cheapest path given the flow. We now further augment the network so that all flow-carrying paths have cost exactly c. One can show that this increases the cost of any path (and thus of any flow in the network) by no more than $\sqrt{\epsilon}$. Now observe that f' is an exact Nash flow in G'. This gives us

$$c_G(f) \leq A(c_{G'}(f^{OPT(G')})) + \epsilon \leq A(c_G(f^{OPT(G)}) + \sqrt{\epsilon}) + \epsilon$$

where A is the price of anarchy in G', $c_N(h)$ denotes the cost of a flow h in a network N, and $f^{OPT(N)}$ denotes the min-cost flow in a network N.

In particular, when all latency functions are linear, we can apply the Roughgarden-Tardos result bounding the price of anarchy by $4/3$ [25].

7 Infinitesimal Users: Multicommodity Flow

In this section, we show how to extend our results to the setting of multicommodity flow. Here, every user is associated with a commodity. Different commodities can have different start and end vertices and even may have access to different subgraphs of the network. This notion of different allowable subgraphs is natural in the context of routing and can model issues such as time-based variations. For example, if one edge represents a given road at 8:30AM and another the same road at 9:30AM, then users who must get to work by 9:00AM would be restricted to the first edge, and the two edges may therefore end up with quite different congestions even at Nash equilibrium. We now summarize the necessary changes to our definitions for this multicommodity setting.

A *commodity* i has an associated start vertex s_i, an end vertex e_i, and an allowed subgraph G_i of G. Let $\mathcal{P}_i$ represent the set of simple s_i-e_i paths on G_i. Let $\mathcal{P}$ represent the union of all $\mathcal{P}_i$. Let f_i be the total amount of commodity i. A *multicommodity flow* is a function $f : \mathcal{P} \rightarrow \mathcal{R}^+$, such that $\sum_i f_i = 1$ and $\sum_{P \in \mathcal{P}_i} f_P = f_i$ for all i.

Similar to Assumption 2.5, we obtain the following multicommodity no-regret assumption:

Assumption 7.1. For every commodity i, the series of flows $f^1, f^2, \ldots$ satisfies

$$\frac{1}{Tf_i} \sum_{t=1}^{T} \sum_{e \in E} \ell_e(f_e^t) f_{e,i}^t \leq R(T) + \frac{1}{T} \min_{P \in \mathcal{P}_i} \sum_{t=1}^{T} \sum_{e \in P} \ell_e(f_e^t)$$

where $R(T) \rightarrow 0$ as T increases and where $f_{e,i}^t$ is the flow of commodity i on edge e at time t.

Summing over these equations, we obtain

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{e \in E} \ell_e(f_e^t) f_e^t \leq R(T) + \frac{1}{T} \sum_i f_i \min_{P \in \mathcal{P}_i} \sum_{t=1}^{T} \sum_{e \in P} \ell_e(f_e^t)$$

As before, the function $R(T)$ may depend on the size of the network and its maximum possible latency, and we define T_ϵ as the number of time steps required to get $R(T) = \epsilon$.

Definition 7.2. A multicommodity flow f is said to be at ϵ-Nash equilibrium if the average cost under this flow is within ϵ of the weighted costs of the minimum cost paths available to each commodity under this flow, i.e. $C(f) - \sum_i f_i \min_{P \in \mathcal{P}_i} \sum_{e \in P} \ell_e(f_e) \leq \epsilon$.

Given these new definitions, the proofs of Theorems 3.1, 4.1, and 5.1 all proceed analogously to yield the following theorems and corollaries:

Theorem 7.3. *In the setting of linear latency functions and multicommodity flow, for $T \geq T_\epsilon$, the average flow $\hat{f}$ is at ϵ-Nash equilibrium.*

Corollary 7.4. *Assume the setting of multicommodity flow and linear latency functions. On general graphs, if all agents use the Kalai-Vempala algorithm [20], the average flow converges to an ϵ-Nash equilibrium at $T_\epsilon = O(\frac{mn \log n}{\epsilon^2})$. On networks consisting of two nodes and m parallel links, if all agents use optimized "combining expert advice"-style algorithms (with each edge an expert), the average flow converges to an ϵ-Nash equilibrium at $T_\epsilon = O(\frac{\log m}{\epsilon^2})$.*

Theorem 7.5. *Let $\epsilon' = \epsilon + 2\sqrt{s\epsilon n}$. Then for general functions with maximum slope s, in the multicommodity flow setting, for $T \geq T_\epsilon$, the average flow is ϵ'-Nash.*

Corollary 7.6. *Assume the multicommodity setting and that all latency functions are positive, non-decreasing, and continuous, with maximum slope s. On general graphs, if all agents use the Kalai-Vempala algorithm [20], the average flow converges to an ϵ'-Nash equilibrium at $T_\epsilon = O(\frac{mn \log n}{\epsilon^2}) = O(\frac{mn^3 s^2 \log n}{\epsilon'^4})$. On networks consisting of two nodes and m parallel links, if all agents use optimized "combining expert advice"-style algorithms, the average flow converges to an ϵ'-Nash equilibrium at $T_\epsilon = O(\frac{\log m}{\epsilon^2}) = O(\frac{n^2 s^2 \log m}{\epsilon'^4})$.*

Theorem 7.7. *In the multicommodity setting, on general graphs with general latency functions, for all but a $(ms^{1/4}\epsilon^{1/4})$ fraction of time steps up to time T_ϵ, f^t is a $(\epsilon + 2\sqrt{s\epsilon n} + 2m^{3/4}s^{1/4}\epsilon^{1/4})$-Nash flow. We can rewrite this as: for all but an ϵ' fraction of time steps up to T_ϵ, f^t is an ϵ'-Nash flow for $\epsilon = \Omega\left(\frac{\epsilon'^4}{sm^4 + s^2 n^2}\right)$.*

Corollary 7.8. *In the multicommodity setting, on general graphs with general latency functions with maximum slope s, for all but a $(ms^{1/4}\epsilon^{1/4})$ fraction of time steps up to time $T = T_\epsilon$, the average cost $\frac{1}{T}\sum_{t=1}^{T} c^t$ incurred by any user is at most $(\epsilon + 2\sqrt{s\epsilon n} + m^{3/4}s^{1/4}\epsilon^{1/4})$ worse than the cost of the best path on that time step.*

Remark 7.9. The price-of-anarchy results sketched in Section 6 also extend to the multicommodity flow setting.

Remark 7.10. In real-world traffic, it would be nonsensical if a user could drive part-way to work on roads that exist at 9:30AM and the rest of the way on the roads that existed at 8:30AM that day. If we wanted to capture the notion that users may have a choice of when to travel, our current model would allow such spurious paths. To avoid this, one could extend our definitions so that each commodity would be associated with a start node, an end node, a subgraph, and a set of allowable hours. Permissible paths for a user would be those consisting only of edges all with the same allowable hour h and all from the allowed subgraph. All the results in this section hold given this further extended definition of commodities.

8 Discrete Users: Parallel Links

In contrast with the previous sections, we now consider discrete users, where we denote the ith user weight as w_i. Without loss of generality, we assume that the weights are normalized such that $\sum_{i=1}^{n} w_i = 1$. We limit ourselves in this section to the single-commodity version of the parallel links model and to functions with latency equal to the load, i.e. for a path e we have $\ell_e = f_e$. For each user i, we let the latency excluding her own path e at time t be $\ell_e(f_e^t \setminus i)$ and her average latency on link e be $\ell_e(\hat{f}_e \setminus i) = \frac{1}{T}\sum_{t=1}^{T} \ell_e(f_e^t \setminus i)$, where $f_e^t \setminus i = f_e^t$ if user i is not routing on link e and $f_e^t \setminus i = f_e^t - w_i$ otherwise. We always exclude the ith player from the latency function, since the ith player always pays for its weight.

Next we observe that at time t, there always exists a path with load at most the average load.

Observation 8.1. At any time step t, for every user i, there exists a path e such that $\ell_e(\hat{f}_e \setminus i) \leq \frac{1-w_i}{m}$.

The following theorem differs from other theorems in the paper in the sense that it is an expectation result and holds for every user.

Theorem 8.2. *Consider the parallel links model, with latency functions such that the latency equals the load. Assume that each discrete user i uses an optimized best expert algorithm. Then for all users, for all $T \geq O(\frac{\log m}{\epsilon^2})$,*

$$E_{e\sim q_t}[\ell_e(f_e^t \setminus i)] \leq \frac{1-w_i}{m} + \epsilon,$$

where q_t is the distribution over the m links output by the best expert algorithm.

Proof. By observation 8.1 we have that there exists a path with average cost at most $\frac{1-w_i}{m}$. Since user i is using an optimized best expert algorithm and the maximal latency is 1, we have that

$$\sum_{t=1}^{T} \mathbf{E}_{e\sim q_t}[\ell_e(f_e^t \setminus i)] \leq \min_{e\in E} \ell_e(\hat{f}_e \setminus i) + \sqrt{\frac{\log m}{T}} \leq \frac{1-w_i}{m} + \sqrt{\frac{\log m}{T}} \leq \frac{1-w_i}{m} + \epsilon,$$

where the last inequality holds for $T \geq O(\frac{\log m}{\epsilon^2})$. □

Consider an instance of this model where every user plays uniformly at random. The resulting flow is clearly a Nash equilibrium, and the expected latency for the ith player is $\frac{1-w_i}{m}$ excluding its own weight. We thus have shown that the expected latency experienced by each user i is at most ϵ worse than this Nash latency.

9 Conclusions

In this paper, we consider the question: if each player in a routing game uses a no-regret strategy, will behavior converge to a Nash equilibrium, and under what conditions and in what sense? Our main result is that in the setting of multicommodity flow and infinitesimal agents, a $1-\epsilon$ fraction of the daily flows are at ϵ-Nash equilibrium for ϵ approaching 0 at a rate that depends polynomially on the players' regret bounds and the maximum slope of any latency function. Moreover, we show the dependence on slope is necessary.

Even for the case of reasonable (bounded) slopes, however, our bounds for general nonlinear latencies are substantially worse than our bounds for the linear case. For instance if agents are running the Kalai-Vempala algorithm [20], we get a bound of $O(\frac{mn\log n}{\epsilon^2})$ on the number of time steps needed for the time-average flow to reach an ϵ-Nash equilibrium in the linear case, but $O(\frac{mn^3\log n}{\epsilon^4})$ for general latencies. We do not know if these bounds in the general case can be improved. In addition, our bounds on the daily flows lose additional polynomial factors which we suspect are not tight.

References

[1] B. Awerbuch and R. Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, 2004.

[2] Y. Azar. *On-line Load Balancing Online Algorithms - The State of the Art*, chapter 8, pages 178–195. Springer, 1998.

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

[4] Petra Berenbrink, Tom Friedetzky, Leslie Ann Goldberg, Paul Goldberg, Zengjian Hu, and Russell Martin. Distributed selfish load balancing. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms(SODA)*, 2006.

[5] N. Cesa-Bianchi, Y. Freund, D.P. Helmbold, D. Haussler, R.E. Schapire, and M.K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

[6] A. Czumaj, P. Krysta, and B. Vöcking. Selfish traffic allocation for server farms. In *Proceedings of the 34th Symposium on Theory of Computing*, pages 287–296, 2002.

[7] A. Czumaj and B. Vöcking. Tight bounds on worse case equilibria. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 413–420, 2002.

[8] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to nash equilibria. In *30th International Conference on Automata, Languages and Programming (ICALP)*, pages 502–513, 2003.

[9] E. Even-Dar and Y. Mansour. Fast convergence of selfish rerouting. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 772–781, 2005.

[10] Alex Fabrikant, Ankur Luthra, Elitza Maneva, Christos H. Papadimitriou, and Scott Shenker. On a network creation game. In *Proceedings of the twenty-second annual symposium on Principles of distributed computing (PODC)*, pages 347–351. ACM Press, 2003.

[11] Simon Fischer, Harald Raecke, and Berthold Vöcking. Fast convergence to wardrop equilibria by adaptive sampling methods. In *Proceedings of 38th ACM Symposium on Theory of Computing (STOC)*, 2006. To appear.

[12] Simon Fischer and Berthold Vöcking. On the evolution of selfish routing. In *Proceedings of the 12th European Symposium on Algorithms (ESA)*, pages 323–334, 2004.

[13] Simon Fischer and Berthold Vöcking. Adaptive routing with stale information. In *Proceedings of the 24th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, 2005.

[14] Dean P. Foster and Rakesh V. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 1997.

[15] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[16] Y. Freund and R. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.

[17] Paul Goldberg. Bounds for the convergence rate of randomized local search in multiplayer games, uniform resource sharing game. In *Proceedings of the Twenty-Third PODC*, pages 131–144, 2004.

[18] J.F. Hannan. Approximation to Bayes risk in repeated play. In M. Dresher, A.W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume III, pages 97–139. Princeton University Press, 1957.

[19] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 2000.

[20] Adam Kalai and Santosh Vempala. Efficient algorithms for on-line optimization. In *Proceedings of the The 16th Annual Conference on Learning Theory*, pages 26–40, 2003.

[21] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *Proceedings of 16th STACS*, pages 404–413, 1999.

[22] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[23] H.B. McMahan and A. Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *Proceedings of the 17th Annual Conference on Learning Theory (COLT)*, pages 109–123, 2004.

[24] I. Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13:111–124, 1996.

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

[26] Tim Roughgarden. On the severity of Braess's paradox: Designing networks for selfish users is hard. In *42nd Annual IEEE Symposium on Foundations of Computer Science*, 2001.

[27] Eiji Takimoto and Manfred K. Warmuth. Path kernels and multiplicative updates. In *Proceedings of the 15th Annual Conference on Computational Learning Theory*, Lecture Notes in Artificial Intelligence. Springer, 2002.

[28] J. G. Wardrop. Some theoretical aspects of road traffic research. In *Proceedings of the Institute of Civil Engineers, Pt. II*, volume 1, pages 325–378, 1952.

[29] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, pages 928–936, 2003.

[30] Martin Zinkevich. Theoretical guarantees for algorithms in multi-agent settings. Technical Report CMU-CS-04-161, Carnegie Mellon University, 2004.