On Learning How Players Learn: Estimation of Learning Dynamics in the Routing Game

Kiet Lam
UC Berkeley
kiet.lam@berkeley.edu

Walid Krichene
UC Berkeley
walid@eecs.berkeley.edu

Alexandre Bayen
UC Berkeley
bayen@berkeley.edu

ABSTRACT

The routing game models congestion in transportation networks, communication networks, and other cyber physical systems in which agents compete for shared resources. We consider an online learning model of player dynamics: at each iteration, every player chooses a route (or a probability distribution over routes, which corresponds to a flow allocation over the physical network), then the joint decision of all players determines the costs of each path, which are then revealed to the players.

We pose the following estimation problem: given a sequence of player decisions and the corresponding costs, we would like to estimate the learning model parameters. We consider in particular entropic mirror descent dynamics, reduce the problem to estimating the learning rates of each player.

We demonstrate this method using data collected from a routing game experiment, played by human participants: We develop a web application to implement the routing game. When players log in, they are assigned an origin and destination on the graph. They can choose, at each iteration, a distribution over their available routes, and each player seeks to minimize her own cost. We collect a data set using this interface, then apply the proposed method to estimate the learning model parameters. We observe in particular that after an exploration phase, the joint decision of the players remains within a small distance of the Nash equilibrium. We also use the estimated model parameters to predict the flow distribution over routes, and compare these predictions to the actual distribution. Finally, we discuss some of the qualitative implications of the experiments, and give directions for future research.

1. INTRODUCTION

The routing game is a non-cooperative game that models congestion in many cyber physical systems (CPS) in which non-cooperative agents compete for shared resources, such as transportation networks (the resources being roads) and communication networks (the resources being communication links) [4, 27, 25]. The game is played on a directed graph that represents the network, and each player is given by a source node and destination node, and seeks to send traffic (either packets in a communication setting, or cars in a transportation setting) while minimizing the total delay of that traffic. The delay is determined by the joint decision of all players, such that whenever an edge has high load, it becomes congested and any traffic using that edge incurs additional delay, defined by a congestion function that models the underlying physical process. This model of congestion is simple yet powerful, and routing games have been studied extensively since the seminal work of Beckman [4].

1.1 Learning models and convergence to Nash equilibria

The Nash equilibria of the game are simple to characterize, and have been used to quantify the inefficiency of the network, using the price of anarchy [27]. However, the Nash equilibrium concept may not offer a good descriptive model of actual behavior of players. Besides the assumption of rationality, which can be questioned [29], the Nash equilibrium assumes that players have a complete description of the structure of the game, their own cost functions, and those of other players. This model is arguably not very realistic for the routing game, as one does not expect users of a network to have an accurate representation of the cost function on every edge of the network, or of the other users of the network. One alternative to the Nash equilibrium concept as a descriptive model of players is a model of repeated play [23, 12], sometimes called learning models [10] or adjustment models [14]. In such models, one assumes that each player makes decisions iteratively, and uses

the outcome of each iteration to adjust their next decision. Formally, if $x_k^{(t)}$ is the decision of player k at iteration t, and $\ell_k^{(t)}$ is the corresponding vector of costs (delays), then player k faces a sequential decision problem in which she iteratively chooses $x_k^{(t)}$ then observes $\ell_k^{(t)}$. These sequential decision problems are coupled through the cost functions, since $\ell_k^{(t)}$ depends not only on $x_k^{(t)}$ but also on $x_{k'}^{(t)}$ for $k' \neq k$ (but players do not necessarily model this coupling). Such models have a long history in game theory, and date back to the work of Hannan [15] and Blackwell [5]. In recent years, there has been a resurgence of research on the topic of learning in games using sequential decision problems, see for example [10] and references therein.

When designing a model of player decisions, many properties are desirable. Perhaps the most important property is that the dynamics should be consistent with the equilibrium of the game, in the following sense: Asymptotically, one should expect the learning dynamics to converge to the equilibrium of the full information, one-shot game (be it Nash equilibrium or other, more general equilibrium concepts). In this sense, players "learn" the equilibrium asymptotically. Much progress has been made in recent years in characterizing classes of learning dynamics which are guaranteed to converge to an equilibrium set [13, 17, 16, 12]. In particular for the routing game, different models of learning have been studied for example in [11, 6, 19, 21, 20], with different convergence guarantees.

1.2 A mirror descent model of learning

We will focus in particular on the mirror descent model used in [22], since it offers a large family of models that have strong convergence guarantees to Nash equilibria. This model describes the learning dynamics as solving, at each step, a simple minimization problem parameterized by a learning rate η. Formally, the decision at iteration $t+1$ is obtained by solving

$$x_k^{(t+1)}(\eta_k^{(t)}) = \arg\min_{x_k \in \Delta^{\mathcal{A}_k}} \eta_k^{(t)} \left\langle \ell_k^{(t)}, x \right\rangle + D_{\psi_k}(x_k, x_k^{(t)}),$$

where ψ_k is a distance generating function with corresponding Bregman divergence D_{ψ_k}, and $\eta_k^{(t)}$ is a learning rate. Intuitively, minimizing the first term $\left\langle \ell_k^{(t)}, x \right\rangle$ will assign traffic to the routes which currently have minimal cost, and minimizing the second term $D_{\psi_k}(x_k, x_k^{(t)})$ will keep the traffic at its current value. Minimizing the linear combination trades-off both terms, and the learning rate $\eta_k^{(t)}$ determines how aggressive the player is in updating her strategy: A small learning rate results in a small change in strategy (i.e. $x_k^{(t+1)}$ is close to $x_k^{(t)}$), while a large learning rate results in a significant change.

1.3 Estimating the learning rates

Motivated by this interpretation of the learning dynamics, we propose the following estimation problem: Given a sequence of player decisions $(x_k^{(t)})$, and the sequence of corresponding costs $(\ell_k^{(t)})$, can we estimate the learning model parameters to fit these observations? These quantities are effectively measured in our experimental setting using the routing interface, and can be measured on transportation networks using existing traffic monitoring and forecasting systems, such as the Mobile Millennium system [2] or the Grenoble Traffic Lab [8].

Our proposed approach is to assume that the player is using a given distance generating function ψ_k, and estimate η_k for example by minimizing the distance between the observed decision $\bar{x}_k^{(t+1)}$, and the decision predicted by the model, $x_k^{(t+1)}(\eta_k^{(t)})$. More precisely, we can choose $\eta_k^{(t)}$ to minimize $D_{\psi_k}(\bar{x}_k^{(t+1)}, x_k^{(t+1)}(\eta_k^{(t)}))$. We show that in the entropic case (when ψ_k is the negative entropy), this problem is convex, thus $\eta_k^{(t)}$ can be estimated efficiently e.g. by using gradient descent. This method allows us to estimate one parameter $\eta_k^{(t)}$ per iteration t and per player k. When we have a sequence of observations available, it can be desirable to control the complexity of the model by assuming a parameterized sequence of learning rates, instead of estimating each term separately. Thus, we propose a second method which assumes that the learning rate is of the form $\eta_k^{(t)} = \eta_k^{(0)} t^{-\alpha_k}$, with $\alpha_k \in (0, 1)$. The resulting estimation problem is non-convex in general, but since it is a two dimensional problem, it can be minimized efficiently. Finally, we consider a family of distance generating functions ψ_ϵ, parameterized by ϵ, that can be viewed as a generalization of the negative entropy function. These generalized entropy functions have desirable properties that will be discussed in more detail.

1.4 Summary of contributions and organization of the article

Our main contributions are to

1. Pose the learning rate estimation problem, and show that it is convex problem in the entropic case. We also give an example application of the estimated model: It can be used to predict the decision of the players over the next few iterations, by propagating the model forward with the estimated values of the parameters.

2. Develop a routing game system in order to collect data on routing decisions. We developed a web interface in which a master user can create an instance of the routing game by defining a graph and cost functions on edges of the graph. Then other users can connect to the interface as players. The game then proceeds similarly to our learning model: At each iteration, every player

chooses a flow distribution on their available routes (using a graphical user interface with sliders), then their decisions are sent to a backend server, which computes the total cost of each route, and sends this information back to each player.

3. Apply the proposed methods to the data collected from the routing game system, and give quantitative and qualitative insights into the decision dynamics of human players. In particular, we observed that in the first few iterations, the flow distributions oscillate, which corresponds to a high value of estimated learning rates. For later iterations, the flow distributions are, in general, close to equilibrium, and the learning rates are lower, although some players may occasionally move the system away from equilibrium by performing an aggressive update (high learning rate). It was also interesting to observe that in some rare cases, the estimate of the learning rate is negative, which means that the player updated her strategy by assigning more traffic to routes with higher cost, a counter-intuitive behavior which is hard to model. Finally, we comment on the performance of the prediction over a short horizon, which seems to indicate that the mirror descent model is a good descriptive model for player behavior in this setting.

The remainder of the article is organized as follows: In Section 2, we formally define the routing game and review the characterization of its equilibria, then define the mirror descent dynamics and review its convergence guarantees. In Section 3, we pose the learning rate estimation problem in the entropy case, then extend it to the generalized entropy case. We also briefly discuss the traffic prediction problem. In Section 4, we describe the experimental setting, some implementation details, and the nature of the collected data. We then use this data to run the estimation and prediction tasks in Section 5, comment on the quality of the prediction, and give some qualitative and quantitative insights into the decision dynamics. We conclude in Section 6 by summarizing our results and giving directions for future research.

2. THE ROUTING GAME AND THE LEARNING MODEL

In this section, we give the definition of the (one-shot) routing game, and the model of learning dynamics.

2.1 The routing game

The routing game is played on a directed graph $\mathcal{G} = (V, E)$, where V is a vertex set and $E \subset V \times V$ is an edge set. The players will be indexed by $k \in \{1, \dots, K\}$, where every player is given by an origin vertex $o_k \in V$, a destination vertex $d_k \in V$, and a traffic mass $m_k \geq 0$ that represents the total traffic that the player needs to send from o_k to d_k. The set of available paths connecting o_k to d_k will be denoted by $\mathcal{P}_k$, and the action set of player k is simply the probability simplex over $\mathcal{P}_k$, which we denote by $\Delta^{\mathcal{P}_k} = \{x \in \mathbb{R}_+^{\mathcal{P}_k} : \sum_{p \in \mathcal{P}_k} x_p = 1\}$. In other words, each player chooses a distribution over their available paths, and their traffic is allocated to paths according to that distribution. We will denote by $x_k \in \Delta^{\mathcal{P}_k}$ the distribution of player k. Note that x_k is a distribution vector, so the vector of actual flows is the scaled vector $m_k x_k$. The joint decision of all players is denoted by $x = (x_1, \dots, x_K)$. The costs of the players are then determined as follows:

a) The cost on an edge e is $c_e(\phi_e(x))$, where $c_e(\cdot)$ is a given, increasing function (this models the actual cost due to the physical process, for example delay on a road segment due to accumulation of cars), and $\phi_e(x)$ is the total traffic flow on edge e induced by the distribution x, obtained simply by summing all the path flows that go through that edge, i.e. $\phi_e(x) = \sum_k \sum_{p \in \mathcal{P}_k} m_k x_{k,p}$.

b) The cost on a path $p \in \mathcal{P}_k$ is denoted by $\ell_{k,p}(x)$, and is the sum of edge costs along the path, i.e. $\ell_{k,p}(x) = \sum_{e \in p} c_e(\phi_e(x))$.

c) The cost for player k is the total path cost for all the traffic sent by player k, i.e. $\sum_{p \in \mathcal{P}_k} m_k x_{k,p} \ell_{k,p}(x)$. This is simply the inner product between the flow vector $m_k x_k$ and the path delay vector $\ell_k(x)$, which we denote by $\langle \ell_k(x), x_k \rangle$.

Remark 1 (A note on the player model) *Some formulations of the routing game, e.g. [28, 21], define the game in terms of* populations *of players, such that each population is an infinite set of players with the same origin and destination. This assumes that each player contributes an infinitesimal amount of flow, so each player can play a single path. In our model, each player is macroscopic, and can split its traffic across multiple routes. Both models are equivalent in terms of analysis, the only difference is the interpretation of the model. We choose the finite player interpretation because it is more consistent with the experimental section of the article, where we run the game with finitely many players.*

Definition 1 (Nash equilibrium) *A distribution $x^\star = (x_1^\star, \dots, x_K^\star)$ is a Nash equilibrium if it satisfies the following condition: For all other feasible distributions $x = (x_1, \dots, x_K)$ and for all k, $\langle \ell_k(x^\star), x_k - x_k^\star \rangle \geq 0$.*

In words, $x^\star$ is a Nash equilibrium if for every player k, the expected cost under $x_k^\star$ is lower than the expected cost under any other distribution x_k. If we define the inner product $\langle x, \ell \rangle = \sum_k \langle x_k, \ell_k \rangle$, then this is equivalent to: $x^\star$ is an equilibrium if and only if $\langle \ell(x^\star), x - x^\star \rangle \geq 0$ for all feasible x. This variational inequality is, in fact, equivalent to the first-order optimality condition of the following potential function, usually referred to as the Rosenthal potential, in reference to [26]:

Proposition 1 (Existence of a convex potential) *Consider a routing game and define the following function* $f(x) = \sum_{e \in E} \int_0^{\phi_e(x)} c_e(u)du$. *Then* f *is convex its gradient is* $\nabla f(x) = \ell(x)$.

This result can be found for example in [27]. Due to the fact that the delay function $\ell(\cdot)$ coincides with the gradient field $\nabla f(\cdot)$ of the Rosenthal potential, the Nash condition can be rewritten as $\langle \nabla f(x^\star), x - x^\star \rangle \geq 0$ for all feasible x, and since f is convex, this is a necessary and sufficient condition for optimality of $x^\star$ (see e.g. Section 4.2.3 in [7]). Therefore the set of Nash equilibria is exactly the set of minimizers of the convex potential f. This is important both for computation (computing a Nash equilibrium can be done by minimizing a convex function), and for modeling: One can model player dynamics as performing a distributed optimization of the potential function. More precisely, if we adopt the point of view presented in the introduction, in which each player faces a sequential decision problem, and plays $x_k^{(t)}$ then observes $\ell_k(x^{(t)})$, then this corresponds to a first-order distributed optimization of the function f, where each player is responsible for updating the variables $x_k^{(t)}$, and observes, at each iteration, the partial gradient $\ell_k(x^{(t)}) = \nabla_{x_k} f(x^{(t)})$. Using this connection to distributed optimization, a model of player dynamics was proposed in [22]. We review the model in the next Section.

2.2 The learning model: Mirror descent dynamics

We will consider the model of distributed learning proposed in [22]. Each player is assumed to perform a mirror descent update given by the following algorithm:

Algorithm 1 Distributed mirror descent dynamics with DGF ψ_k and learning rates $(\eta_k^{(t)})$.

1: **for** each iteration $t \in \{1, 2, \dots\}$ **do**
2: **for** each player $k \in \{1, \dots, K\}$ **do**
3: Play $x_k^{(t)}$,
4: Observe $\ell_k^{(t)} = \nabla_{x_k} f(x^{(t)})$,
5: Update distribution

$$x_k^{(t+1)} = \underset{x_k \in \Delta^{\mathcal{P}_k}}{\arg\min} \left[\eta_k^{(t)} \left\langle \ell_k(x^{(t)}), x_k \right\rangle + D_{\psi_k}(x_k, x_k^{(t)}) \right] \quad (1)$$

In the update equation (1), $D_{\psi_k}(x_k, x_k^{(t)})$ is the Bregman divergence between the distributions x_k and $x_k^{(t)}$, defined as $D_\psi(x, y) = \psi(x) - \psi(y) - \langle \nabla\psi(y), x - y \rangle$, for a strongly convex function ψ, called the distance generating function (DGF), see for example [9, 1] for a review of Bregman divergences and their uses in optimization. Some special cases include:

a) The Euclidean case: If $\psi(x) = \frac{\|x\|_2^2}{2}$, then $D_\psi(x, y) = \frac{\|x-y\|_2^2}{2}$. In this case, mirror descent reduces to the projected gradient descent algorithm.

b) The entropic case: If $\psi(x) = -H(x)$ where $H(x) = -\sum_p x_p \ln x_p$ is the negative entropy, then $D_\psi(x, y) = \sum_p x_p \ln \frac{x_p}{y_p}$ is the Kullback-Leibler (KL) divergence from x to y. In this case, the mirror descent algorithm is sometimes called the entropic descent [3], or exponentiated gradient descent [18].

The mirror descent method is a general method for convex optimization proposed in [24]. The model in Algorithm 1 is a distributed version of mirror descent, applied to the potential function f (defined in Proposition 1). To give some intuition of the method, the first term $\left\langle \ell_k^{(t)}, x_k \right\rangle$ in the minimization problem (1) can be thought of as a linear approximation of the potential function (since $\ell(x) = \nabla f(x)$), and the second term $D_\psi(x_k, x_k^{(t)})$ penalizes deviations from the previous iterate $x_k^{(t)}$. The learning rate $\eta_k^{(t)}$ determines the tradeoff between the two terms, and can be thought of as a generalized step size: A smaller $\eta_k^{(t)}$ results in a distribution which is closer to the current $x_k^{(t)}$. Thus, from the potential function point of view, the player minimizes a linearization of the potential plus a Bregman divergence term that keeps x_k close to $x_k^{(t)}$. From the routing game point of view, the first term $\left\langle \ell_k^{(t)}, x_k \right\rangle$ corresponds to putting weight on the paths that have smaller cost during the previous iteration, and the second term keeps the distribution close to its current value. The learning rate parameter $\eta_k^{(t)}$ determines how aggressive the player is in shifting traffic to the paths which appear to be the best.

The convergence of this distributed learning model is discussed in [22]. The learning dynamics given in Algorithm 1 is guaranteed to converge under the following assumptions:

Theorem 1 (Theorem 3 in [22]) *Consider the routing game with mirror descent dynamics defined in Algorithm 1, and suppose that for all* k, $\eta_k^{(t)}$ *is decreasing to* 0. *Then* $f(x^{(t)}) - f(x^\star) = \mathcal{O}\left(\sum_k \frac{1}{t\eta_k^{(t)}} + \frac{\sum_{\tau=1}^t \eta_k^{(\tau)}}{t} \right)$.

In particular, if $\eta_k^{(t)} = \eta_k^{(0)} t^{-\alpha_k}$, with $\alpha_k \in (0, 1)$, then one can bound the sum $\sum_{\tau=1}^t \eta_k^{(t)} = \eta_k^{(0)} \sum_{\tau=1}^t \tau^{-\alpha_k} \leq \eta_k^{(0)} \int_0^t \tau^{-\alpha_k} d\tau = \frac{\eta_k^{(0)}}{1-\alpha_k} t^{1-\alpha_k}$. Therefore, $f(x^{(t)}) - f(x^\star) = \mathcal{O}(t^{\alpha_k - 1}) + \mathcal{O}(t^{-\alpha_k}) = \mathcal{O}(t^{-\min(\alpha_k, 1-\alpha_k)})$, which converges to 0. While this specific convergence rate does not matter for the purposes of the estimation problem, the convergence guarantees for decaying learning rates motivates the modeling assumptions made in the next section.

3. LEARNING MODEL ESTIMATION

In this section, we assume that we have access to a sequence of observations of traffic distributions $(\bar{x}_k^{(t)})$, and a sequence of delay vectors $(\bar{\ell}_k^{(t)})$, for a given player k. The over bar is used to make a clear distinction between quantities which are observed (e.g. $\bar{x}_k^{(t)}$) and quantities which are estimated or predicted (e.g. $x_k^{(t)}$). Given this sequence of observations, we would like to fit a model of learning dynamics. From the previous section, the learning model in Algorithm 1 is naturally parameterized by the DGF ψ_k and the learning rate sequence $(\eta_k^{(t)})$. We will assume that the DGF is given, and discuss how one can estimate the learning rates.

3.1 Estimating a single term of the learning rates sequence

Given the current flow distribution $\bar{x}_k^{(t)}$ and the current delay vector $\bar{\ell}_k^{(t)}$, the mirror descent model prescribes that the next distribution is given by

$$x_k^{(t+1)}(\eta) = \arg\min_{x \in \Delta^{\mathcal{P}_k}} \eta \left\langle \bar{\ell}_k^{(t)}, x_k \right\rangle + D_{\psi_k}(x_k, \bar{x}_k^{(t)}), \quad (2)$$

where ψ_k is given. Therefore, $x_k^{(t+1)}$ can be viewed as a function of η, (hence the notation $x_k^{(t+1)}(\eta)$) and to estimate η, one can minimize

$$d_k^{(t)}(\eta) = D_{\psi_k}(\bar{x}_k^{(t+1)}, x_k^{(t+1)}(\eta)).$$

The problem is then simply

$$\eta_k^{(t)} = \arg\min_{\eta \geq 0} d_k^{(t)}(\eta). \quad (3)$$

In the next proposition, we show that this problem is convex when the DGF is the negative entropy. In fact, one can explicitly compute the gradient of $d_k(\eta)$ in this case, which makes it possible to solve Problem (3) efficiently using gradient descent for example.

Theorem 2 *If ψ_k is the negative entropy, then $d_k^{(t)}(\eta) = D_{\psi_k}(\bar{x}_k^{(t+1)}, x_k^{(t+1)}(\eta))$ is a convex function of η, and its gradient with respect to η is given by*

$$\frac{d}{d\eta} d_k^{(t)}(\eta) = \left\langle \bar{\ell}_k^{(t)}, \bar{x}_k^{(t+1)} - x_k^{(t+1)}(\eta) \right\rangle.$$

PROOF. When ψ_k is the negative entropy, the solution of the mirror descent update (2) can be computed in closed form, and is given by

$$x_{k,p}^{(t+1)}(\eta) = \frac{\bar{x}_{k,p}^{(t)} e^{-\eta \bar{\ell}_{k,p}^{(t)}}}{Z_k^{(t)}(\eta)} \quad (4)$$

where $Z_k^{(t)}(\eta)$ is the appropriate normalization constant, given by $Z_k^{(t)}(\eta) = \sum_p \bar{x}_{k,p}^{(t)} e^{-\eta \bar{\ell}_{k,p}^{(t)}}$, see for example [3] for a proof of this result. Given this expression of $x_k^{(t+1)}(\eta)$, we can explicitly compute the Bregman divergence (which, in this case, is the KL divergence):

$$\begin{aligned} d_k(\eta) &= D_{KL}(\bar{x}_k^{(t+1)}, x_k^{(t+1)}(\eta)) \\ &= \sum_{p \in \mathcal{P}_k} \bar{x}_{k,p}^{(t+1)} \ln \frac{\bar{x}_{k,p}^{(t+1)}}{x_{k,p}^{(t+1)}(\eta)} \\ &= \sum_{p \in \mathcal{P}_k} \bar{x}_{k,p}^{(t+1)} \left(\ln \frac{\bar{x}_{k,p}^{(t+1)}}{\bar{x}_{k,p}^{(t)}} + \eta \bar{\ell}_{k,p}^{(t)} + \ln Z_k^{(t)}(\eta) \right) \\ &= D_{KL}(\bar{x}_k^{(t+1)}, \bar{x}_k^{(t)}) + \eta \left\langle \bar{\ell}_k^{(t)}, \bar{x}_k^{(t+1)} \right\rangle + \ln Z_k^{(t)}(\eta), \end{aligned}$$

where we used the explicit form (4) of $x_k^{(t+1)}(\eta)$ in the third equality, and the fact that $\sum_p \bar{x}_{k,p}^{(t+1)} = 1$ in the last equality. In this expression, the first term does not depend on η, the second term is linear in η, and the last term is the function $\eta \mapsto \ln Z_k^{(t)}(\eta) = \ln \sum_p \bar{x}_{k,p}^{(t)} e^{-\eta \bar{\ell}_{k,p}^{(t)}}$, which is known to be convex in η (see for example Section 3.1.5 in [7]). Therefore $d_k^{(t)}(\eta)$ is convex, and its gradient can be obtained by differentiating each term

$$\begin{aligned} \frac{d}{d\eta} d_k^{(t)}(\eta) &= \left\langle \bar{\ell}_k^{(t)}, \bar{x}_k^{(t+1)} \right\rangle + \frac{\frac{d}{d\eta} Z_k^{(t)}(\eta)}{Z_k^{(t)}(\eta)} \\ &= \left\langle \bar{\ell}_k^{(t)}, \bar{x}_k^{(t+1)} \right\rangle + \frac{\sum_p -\bar{\ell}_{k,p}^{(t)} \bar{x}_{k,p}^{(t)} e^{-\eta \bar{\ell}_{k,p}^{(t)}}}{Z_k^{(t)}(\eta)} \\ &= \left\langle \bar{\ell}_k^{(t)}, \bar{x}_k^{(t+1)} \right\rangle - \left\langle \bar{\ell}_k^{(t)}, x_k^{(t+1)}(\eta) \right\rangle, \end{aligned}$$

which proves the claim. □

While we cannot prove that the problem is convex in the general case (when ψ_k is any DGF), since the problem is one-dimensional, one can apply any non-convex optimization method, such as simulated annealing, to find a local optimum of $d_k^{(t)}(\eta)$.

3.2 Estimating the decay rate of the learning rate sequence

In the previous section, we proposed a method to estimate one term of the learning rate sequence. One can of course repeat this procedure at every iteration, thus generating a sequence of estimated learning rates. However, the resulting sequence may not be decreasing. In order to be consistent with the assumptions of the model, we can assume a parameterized sequence of learning rates (which is by construction decreasing), then estimate the parameters of the sequence, given the observations. Motivated by Theorem 1, we will assume, in this section, that $\eta_k^{(t)} = \eta_k^{(0)} t^{-\alpha_k}$ with $\eta_k^{(0)} > 0$ and $\alpha_k \in (0, 1)$.

Given the observations $(\bar{x}_k^{(t)})$ and $(\bar{\ell}_k^{(t)})$, we can define

a cumulative cost,

$$D_k^{(t)}(\alpha_k, \eta_k^{(0)}) = \sum_{\tau=1}^{t} d_k^{(\tau)}(\eta_k^{(0)} \tau^{-\alpha_k}),$$

then estimate $(\alpha_k, \eta_k^{(0)})$ by solving the problem

$$(\alpha_k, \eta_k^{(0)}) = \underset{\alpha_k \in (0,1), \eta^{(0)} \geq 0}{\arg\min} D_k^{(t)}(\alpha, \eta^{(0)}). \quad (5)$$

Note that this problem is non-convex in general, however, since it is low-dimensional (two parameters to estimate), it can also be solved efficiently using non-convex optimization techniques.

3.3 A parameterized family of distance generating functions

In this section, we propose to use a generalization of the entropy DGF, motivated by the following observation: according to the entropy update and its explicit solution (5), the support of $x_k^{(t+1)}(\eta)$ always coincides with the support of $\bar{x}_k^{(t)}$ (due to the multiplicative form of the solution). As a consequence, if we observe two consecutive terms $\bar{x}_k^{(t)}, \bar{x}_k^{(t+1)}$ such that some p is in the support of $\bar{x}_k^{(t+1)}$ but not in the support of $\bar{x}_k^{(t)}$, the KL divergence $D_{KL}(\bar{x}_k^{(t+1)}, x_k^{(t+1)}(\eta))$ is infinite for all η, since $\text{support}(\bar{x}^{(t+1)}) \not\subset \text{support}(x_k^{(t+1)}(\eta))$ (in measure theoretic terms, $\bar{x}^{(t+1)}$ is not absolutely continuous with respect to $x_k^{(t+1)}(\eta)$). This is problematic, as the estimation problem is ill-posed in such cases (which do occur in the data set used in Section 5). To solve this problem, we consider the following DGF: For $\epsilon > 0$, let

$$\psi_\epsilon(x_k) = -H(x + \epsilon) = \sum_p (x_{k,p} + \epsilon) \ln(x_{k,p} + \epsilon).$$

The corresponding Bregman divergence is

$$D_{\psi_\epsilon}(x_k, y_k) = \sum_p (x_{k,p} + \epsilon) \ln \frac{x_{k,p} + \epsilon}{y_{k,p} + \epsilon},$$

and can be interpreted as a generalized KL divergence. In particular, for any $\epsilon > 0$, this Bregman divergence is finite for any $x_k, y_k \in \Delta^{\mathcal{P}_k}$, unlike the KL divergence. Additionally, the support is not necessarily preserved. Finally, it is worth observing that when $\epsilon > 0$, the update equation (1) does not have a closed-form expression as in (4). In our numerical simulations in Section 5, we use the generalized entropy DGF proposed here.

3.4 Traffic flow prediction

We discuss one important application of the proposed estimation problem. Once we have estimated the learning rates, we can propagate the model forward in order to predict the distributions of the players for the next time step. More precisely, if at iteration t, we have observed $\bar{x}^{(t)}$, $\bar{\ell}^{(t)}$, and we have estimated the terms $(\eta_k^{(1)}, \ldots, \eta_k^{(t-1)})$ for a player k, then we can use these terms to estimate $\eta_k^{(t)}$, and predict the next distribution by solving

$$\begin{aligned} x_k^{(t+1)} &= \arg\min x_k \in \Delta^{\mathcal{P}_k} \left\langle \eta_k^{(t)}, \ell(\bar{x}_k^{(t)}) \right\rangle + D_{\psi_k}(x_k, \bar{x}_k^{(t)}) \\ &\triangleq g(\bar{x}_k^{(t)}, \eta_k^{(t)}), \end{aligned}$$

where we defined the function g, which takes a distribution and a learning rate and propagates the model forward one step. We can inductively estimate the next terms by propagating the model further over a horizon h: let $x_k^{(t)} = \bar{x}_k^{(t)}$ and for $i \in \{0, \ldots, h-1\}$,

$$x_k^{(t+i+1)} = g(x_k^{(t+i)}, \eta_k^{(t+i)}). \quad (6)$$

Here, we assume that we can extrapolate the learning rate sequence to estimate the terms $\eta_k^{(t+i)}$. If we assume a particular form of the sequence, $\eta_k^{(t)} = \eta_k^{(0)} t^{-\alpha_k}$, then this can be done readily once we have an estimate of $\eta_k^{(0)}$ and α_k. However, if each term of the sequence is estimated separately, we need to use a model to predict the next terms. We propose these simple methods that are evaluated in Section 5:

1. First, as baseline method, we set $\eta_k^{(t+i)} = \eta_k^{(t-1)}$ for all i (we use the last estimated value).
2. Second, we set $\eta_k^{(t+i)} = \frac{1}{N} \sum_{n=1}^{N} \eta_k^{(t-n)}$ for all i (we use the mean of the last N values).
3. Third, we assume a polynomial decay of the form $\eta_k^{(t)} = \eta_k^{(0)} t^{-\alpha_k}$, and estimate $\alpha_k, \eta_k^{(0)}$ by performing a linear regression on $(\ln \tau, \ln \eta_k^{(\tau)})$ (since the model is linear in logarithmic scale).

We conclude this section by observing that while we chose to apply the model to a simple prediction task, the estimated model can be used, more generally, in any a receding-horizon optimal control problem, by using the current estimate of the model as a plant in the control problem.

4. THE ROUTING GAME WEB APPLICATION

We developed a web application that implements the repeated routing game described in Section 2. The general architecture of the system is summarized in Figure 2. It consists of a client interface that is used by human participants, shown in Figure 1, and a backend server that is responsible for collecting inputs from the clients, updating the state of the game, then broadcasting current information to each player.

A root user can set up the game by creating a graph and defining the cost functions on each edge. Then once a game is set up, players can log in to the client interface, and each player is assigned an arbitrary origin node and

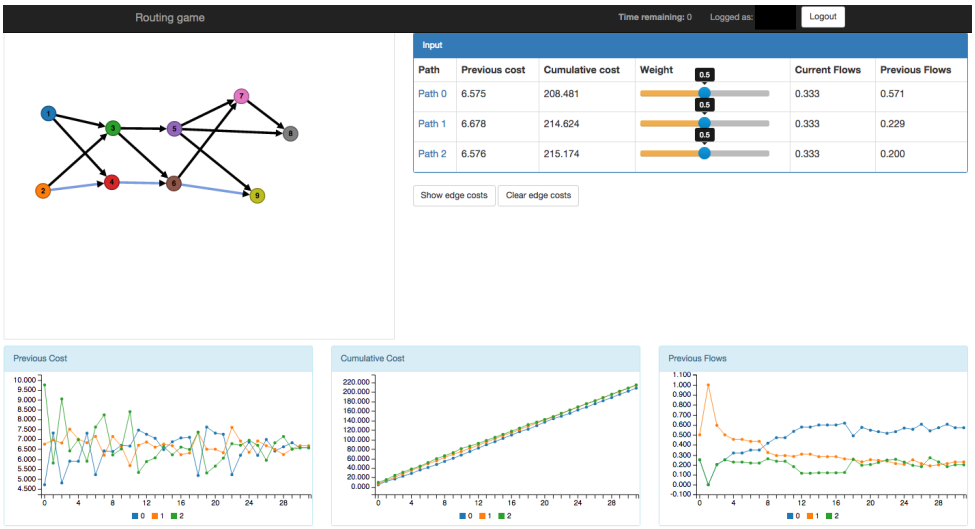

Figure 1: Screenshot of the client side of the routing game application. The table is the main interface on the client side, and can be used by the player to set weights on the different paths, using the sliders. The weights determine the flow distribution $\bar{x}_k^{(t+1)}$. The table also show the previous flows $(x_k^{(t)})$, the previous costs $(\bar{\ell}_k^{(t)})$, and the cumulative costs $\bar{L}_k^{(t)} = \sum_{\tau \le t} \bar{\ell}_k^{(\tau)}$. Clicking a path will also highlight the path on the graph. The bottom charts show the full history of flows, costs, and cumulative losses.

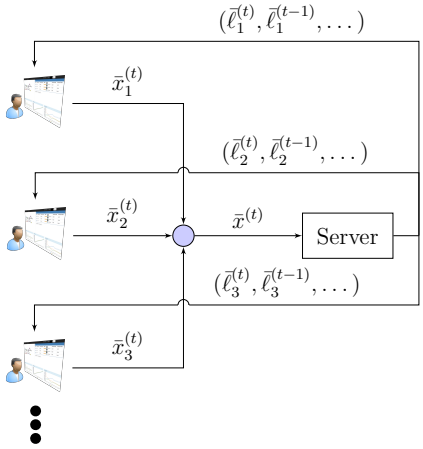

Figure 2: General architecture of the system. During iteration t, the clients input the current values of the distributions $\bar{x}_k^{(t)}$ and send them to the server. At the end of the iteration, the server uses these values to compute the cost functions $\bar{\ell}_k^{(t)}$ and sends them back to the clients.

destination node on the graph. Once the game starts, it is played in iterations, such that each iteration lasts a specified period of time shown by the timer on top of the client interface (each iteration lasts 30 seconds in our experiments). Each player k can use the sliders to set her flow distribution $x_k^{(t)}$ during iteration t. At the end of the iteration, the server uses the values of $x_k^{(t)}$ for all players $k \in \{1, \dots, K\}$ to compute the cost functions $\ell_k^{(t)}$, then sends this information to the client side, which then updates the charts and the table with the last value of the cost. Note that client k only has access to the information about player k, so in this sense, the learning is completely distributed, as players do not observe the decision or the costs of other players. The decisions of the players $(\bar{x}_k^{(t)})$ and the costs $(\bar{\ell}_k^{(t)})$ are logged by the server, with no additional identifiable information about the players.

The code for the web application is available on Github at the following url: `github.com/kietdlam/routing`. To illustrate the methods proposed in this article, we ran the experiment on a small network (shown in the interface in Figure 1), with 5 players. The numerical results are discussed in the next section.

5. EXPERIMENTAL RESULTS

We use the data set collected by the experiment to illustrate the estimation and prediction problems proposed in Section 3, and give some comments on the decision dynamics of the players.

5.1 Distance to equilibrium

First, we evaluate whether the (distributed) decisions of the players converges to the Nash equilibrium of the game. The distance to equilibrium can be measured simply by the Rosenthal potential defined in Proposition 1. Figure 3 shows the potential $f(x^{(t)}) - f(x^\star)$ as a function of iteration t, as well as the corresponding

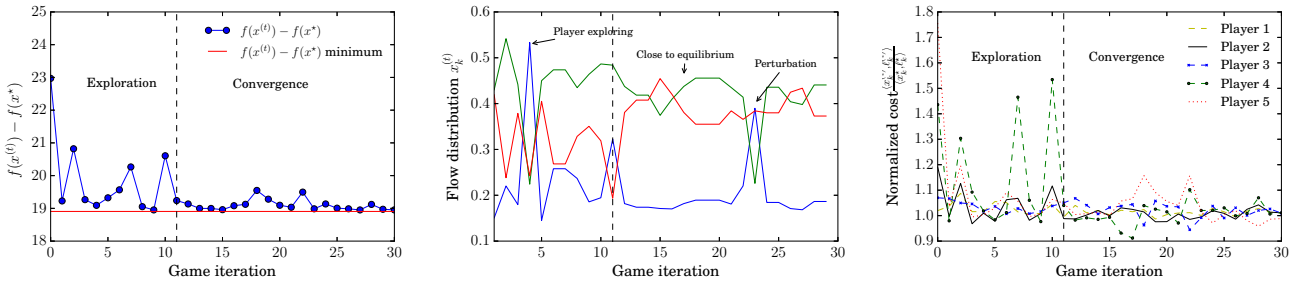

Figure 3: Exploration and convergence to equilibrium. The left figure shows the distance to equilibrium, measured by the Rosenthal potential $f(x^{(t)}) - f(x^\star)$ as a function of iteration t, where $x^{(t)} = (x_1^{(t)}, \ldots, x_K^{(t)})$ is the joint decision of all players. The middle figure shows the flow distribution for a given player, and the right plot shows the costs of all players, normalized by the equilibrium costs (so that their values are comparable).

player costs $\left\langle \ell_k^{(t)}, x_k^{(t)} \right\rangle$ of the players. We can observe that at the beginning of the game, there is a clear exploration phase in which players tend to make aggressive adjustments in their distributions, while during later turns, the adjustments become less aggressive and the joint distribution $x^{(t)}$ remains close to equilibrium (as measured by the potential function f). The system does move away from equilibrium at some later turns (due to a player performing an aggressive update, see for example turn 22 in Figure 3), but it quickly recovers in general.

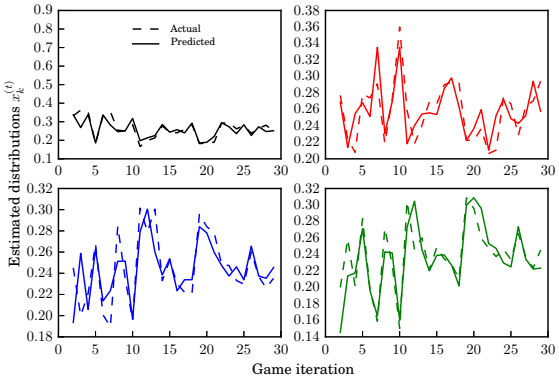

Figure 4: Comparison of the distributions $x_k^{(t)}$ of the estimated model to the actual distributions $\bar{x}_k^{(t)}$, for player $k = 2$. Each subplot corresponds to a path.

5.2 Estimation and prediction

We now apply the method proposed in Section 3 to estimate the learning rates of each player, then use the estimated rates to predict the decision of the players over a short horizon. In this section, we take the Bregman divergence to be the regularized entropy defined in Section 3.3, with $\epsilon = 10^{-3}$.

First, we solve Problem (3) to estimate the learning rate sequence one term at a time. Figure 4 compares the estimated distributions by the model, to the actual distributions. This shows that choosing one value of $\eta_k^{(t)}$ per turn makes it possible to closely fit the observations.

Then, we use the parameterized form $\eta_k^{(t)} = \eta_k^{(0)} t^{-\alpha_k}$, and estimate $\eta_k^{(0)}$ and $\alpha_k^{(0)}$ by solving the problem 5. The results of both methods are shown in Figure (5) When we estimate one term at a time, the resulting sequences have very large variations, and to better visualize them, we plot a moving average (over a window of 5 iterations).

It was interesting and perhaps surprising to observe that when estimating learning rates one term at a time, in some rare instances, the objective $d_k^{(t)}(\eta_k^{(t)})$ is minimal at a negative $\eta_k^{(t)}$, which means that the player shifted the probability mass towards paths with *higher* costs. One such example is given in the table below.

Path	$\bar{x}^{(t)}$	$\bar{\ell}^{(t)}$	$\bar{x}^{(t+1)}$
p_1	.198	6.455	.213
p_2	.218	6.037	.240
p_3	.280	5.933	.301
p_4	.304	6.055	.246

Table 1: Example of an irrational behavior (corresponding to iteration $t = 12$ for player P2) which is hard to predict by the model. The inner product $\left\langle \bar{\ell}_k^{(t)}, \bar{x}_k^{(t+1)} - \bar{x}_k^{(t)} \right\rangle > 0$, which means that the player shifts probability mass to paths with higher costs (in particular, the flow on path p_1 increased even though this is the worst path).

Next, we use the estimated learning rates to predict the distributions of the players over a short horizon $h \in \{1, \ldots, 8\}$. More precisely, given a horizon h, we compute, at each iteration t, the estimated learning rates up to t, then propagate the model forward from t

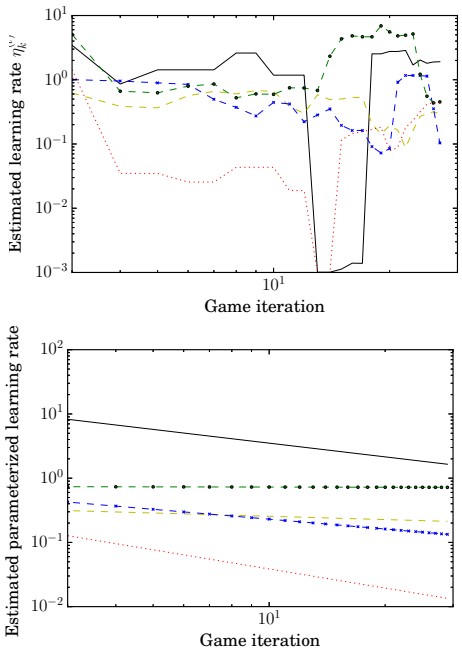

Figure 5: Estimated sequences of learning rates in logarithmic scale. In the top figure, we estimate one term of the sequence at a time then plot a moving average with a window length equal to 5. In the bottom figure, we estimate for each player the initial term $\eta_k^{(0)}$ and the decay rate α_k.

to $t+h$, by iteratively applying the function g defined in (6).

We evaluate each method by computing the average Bregman divergence (per player and per iteration) between the predicted distribution $x_k^{(t+h)}$ and the actual distribution $\bar{x}_k^{(t+h)}$,

$$\frac{1}{K}\sum_{k=1}^{K}\frac{1}{t_{\max}-t_{\min}}\sum_{t=t_{\min}}^{t_{\max}-1} D_{\psi_k}(\bar{x}_h^{(t+h)}, x_k^{(t+h)}),$$

where $t_{\min}$ is taken to be equal to 5 (so that there is always a minimal history of observations to estimate the parameters). The results are given in Figure 6. One can observe that for all methods, as the horizon h increases, the average divergence increases, since the modeling errors propagate and the quality of our predictions degrade. The best overall performance is obtained with the parameterized model $\eta_k^{(t)} = \eta_k^{(0)} t^{-\alpha_k}$, although for $h = 1$, the best prediction is achieved using the per-iteration estimate of $\eta_k^{(t)}$.

6. CONCLUSION

We proposed a problem of model estimation in the routing game, to fit a distributed learning model to sequential observations of player decisions. The estimated

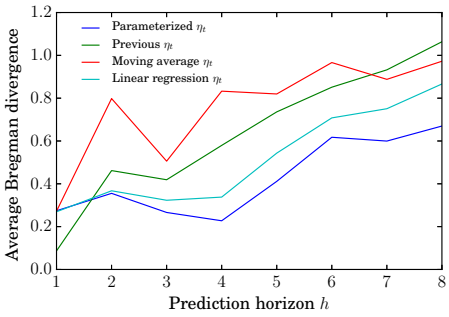

Figure 6: Average Bregman divergence per player and per iteration, between the predicted distributions and the actual distributions, as a function of the prediction horizon.

model can then be used to predict the decisions at future iterations, or, more generally, as a plant model in an optimal control problem.

We considered in particular a model based on the mirror descent algorithm, parameterized by a DGF ψ_k and a sequence of learning rates $(\eta_k^{(t)})$, and gave an intuitive interpretation of how this model can describe player behavior. We showed that the problem of estimating one term of the learning rate sequence is convex in the case of the KL divergence (it remains open to prove this result for other Bregman divergences). To control the complexity of the model and to make the estimation consistent with the theoretical assumptions (decreasing learning rates), we proposed to parameterize the sequence with an initial term $\eta_k^{(0)}$ and a decay rate $\alpha_k \in (0,1)$. When we tested these methods on data collected from our routing game interface, the parameterized sequence estimation outperformed the other methods on the prediction task. Our test results suggest that the mirror descent model can be a good descriptive model of player behavior, although in some rare cases, a player decision can be hard to model (e.g. when a player increase traffic assignment on previously bad routes).

This estimation problem can be extended in several ways: First, in our method, we fixed the DGF to be the negative entropy (regularized in order to avoid situations in which the estimation problem is ill-posed). One could also estimate the DGF itself, in addition to estimating the learning rates. One natural way to pose the estimation problem is to consider a finite collection of distance generating functions $\{\psi_i\}_{i\in\mathcal{I}}$, then to assume that each player k uses a linear combination with weights θ_k $\psi = \sum_i \theta_{k,i}\psi_i$, then estimate the parameter vector θ_k.

7. REFERENCES

[1] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with bregman divergences. *J. Mach. Learn. Res.*, 6:1705–1749, Dec. 2005.

[2] A. Bayen, J. Butler, A. Patire, CCIT, UC Berkeley ITS, and California Dpartment of Transportation, Division of Research and Innovation. *Mobile Millennium Final Report*. 2011.

[3] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3):167–175, May 2003.

[4] M. J. Beckmann, C. B. McGuire, and C. B. Winsten. Studies in the economics of transportation. 1955.

[5] D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956.

[6] A. Blum, E. Even-Dar, and K. Ligett. Routing without regret: on convergence to nash equilibria of regret-minimizing algorithms in routing games. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, PODC '06, pages 45–52, New York, NY, USA, 2006. ACM.

[7] S. Boyd and L. Vandenberghe. *Convex Optimization*, volume 25. Cambridge University Press, 2010.

[8] C. Canudas De Wit, F. Morbidi, L. Leon Ojeda, A. Y. Kibangou, I. Bellicot, and P. Bellemain. Grenoble Traffic Lab: An experimental platform for advanced traffic monitoring and forecasting. *IEEE Control Systems*, 35(3):23–39, June 2015.

[9] Y. Censor and S. Zenios. *Parallel Optimization: Theory, Algorithms and Applications*. Oxford University Press, 1997.

[10] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[11] S. Fischer and B. Vöcking. On the evolution of selfish routing. In *Algorithms–ESA 2004*, pages 323–334. Springer, 2004.

[12] M. J. Fox and J. S. Shamma. Population games, stable games, and passivity. *Games*, 4(4):561–583, 2013.

[13] Y. Freund and R. E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1):79–103, 1999.

[14] D. Fudenberg and D. K. Levine. *The theory of learning in games*, volume 2. MIT press, 1998.

[15] J. Hannan. Approximation to Bayes risk in repeated plays. *Contributions to the Theory of Games*, 3:97–139, 1957.

[16] S. Hart. Adaptive heuristics. *Econometrica*, 73(5):1401–1430, 2005.

[17] S. Hart and A. Mas-Colell. A general class of adaptive strategies. *Journal of Economic Theory*, 98(1):26 – 54, 2001.

[18] J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1 – 63, 1997.

[19] R. Kleinberg, G. Piliouras, and E. Tardos. Multiplicative updates outperform generic no-regret learning in congestion games. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 533–542. ACM, 2009.

[20] S. Krichene, W. Krichene, R. Dong, and A. Bayen. Convergence of heterogeneous distributed learning in stochastic routing games. In *53rd Allerton Conference on Communication, Control and Computing*, 2015.

[21] W. Krichene, B. Drighès, and A. Bayen. Learning nash equilibria in congestion games. *SIAM Journal on Control and Optimization (SICON)*, 2015.

[22] W. Krichene, S. Krichene, and A. Bayen. Convergence of mirror descent dynamics in the routing game. In *European Control Conference (ECC)*, 2015.

[23] J. Marden and J. Shamma. Game theory and distributed control. In H. Young and S. Zamir, editors, *Handbook of Game Theory Vol. 4*. Elsevier Science, 2013.

[24] A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience series in discrete mathematics. Wiley, 1983.

[25] A. Ozdaglar and R. Srikant. Incentives and pricing in communication networks. *Algorithmic Game Theory*, pages 571–591, 2007.

[26] R. W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

[27] T. Roughgarden. Routing games. In *Algorithmic game theory*, chapter 18, pages 461–486. Cambridge University Press, 2007.

[28] W. H. Sandholm. Potential games with continuous player sets. *Journal of Economic Theory*, 97(1):81–108, 2001.

[29] H. A. Simon. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69(1):pp. 99–118, 1955.